PTO/AIA/50 (09-14)
Approved for use through 10/31/2016. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no persons are required to respond to a collection of information unless it displays a valid OMB control number

REISSUE PATENT APPLICATION TRANSMITTAL

Address to:

Mail Stop Reissue
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attorney Docket No.	NRLA-001/03US
First Named Inventor	Anatoli Gorchetchnikov
Original Patent Number	9,189,828
Original Patent Issue Date (Month/Day/Year)	11/17/2015
Express Mail Label No.	

APPLICATION FOR REISSUE OF: (*Check applicable box*) ☑ Utility Patent ☐ Design Patent ☐ Plant Patent

APPLICATION ELEMENTS (37 CFR 1.173)

1. ☐ **Fee Transmittal Form** (PTO/SB/56)
2. ☑ **Applicant asserts small entity status.** See 37 CFR 1.27
3. ☐ **Applicant certifies micro entity status.** See 37 CFR 1.29. Applicant must attach form PTO/SB/15A or B or equivalent.
4. ☑ **Specification and Claims** in double column copy of patent format (*amended, if appropriate*)
5. ☑ **Drawing(s)** (*proposed amendments, if appropriate*)
6. ☑ **Reissue Oath/Declaration or Substitute Statement** (37 CFR 1.175) (PTO/AIA/05, 06, or 07)
7. ☑ **Application Data Sheet** **NOTE:** Benefit claims under 37 CFR 1.78 and foreign priority claims under 37 CFR 1.55 MUST be set forth in an Application Data Sheet (ADS).
8. ☑ **Original U.S. Patent currently assigned?** ☑ Yes ☐ No (*If Yes, check applicable box(es)*)
   - ☑ Written Consent of all Assignees (PTO/AIA/53)
   - ☑ 37 CFR 3.73(c) Statement (PTO/AIA/96)
9. ☐ **CD-ROM or CD-R** in duplicate, Computer Program (*Appendix*) or large table
   - ☐ Landscape Table on CD
10. **Nucleotide and/or Amino Acid Sequence Submission** (*if applicable, items a. – c. are required*)
    - a. ☐ Computer Readable Form (CRF)
    - b. ☐ Specification Sequence Listing on:
      - i. ☐ CD-ROM (2 copies) or CD-R (2 copies); **or**
      - ii. ☐ Paper
    - c. ☐ Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

11. ☑ **Statement of status and support for all changes to the claims.** See 37 CFR 1.173(c).
12. ☑ **Power of Attorney**
13. ☐ **Information Disclosure Statement (IDS)** PTOSB/08 or PTO-1449
    - ☐ Copies of citations attached
14. ☐ **English translation of Reissue Oath/Declaration** (*if applicable*)
15. ☐ **Return Receipt Postcard** (MPEP § 503) (Should be specifically itemized)
16. ☑ **Preliminary Amendment** (37 CFR 1.173; MPEP § 1453)
17. ☑ **Other:** The statement of status and support for all changes to the claims appears in the "Remarks" section of the Preliminary Amendment.

☐ This is a continuation reissue or divisional reissue application (*i.e.*, a second or subsequent reissue application for the same issued patent). (*Check box if applicable.*)

18. CORRESPONDENCE ADDRESS

☑ The address associated with Customer Number: 58249 **OR** ☐ Correspondence address below

Name					
Address					
City		State		Zip Code	
Country		Telephone			
Email					

Signature	/Christopher Max Colice/	Date	November 9, 2016
Name (Print/Type)	Christopher Max Colice	Registration No.	65,634

This collection of information is required by 37 CFR 1.173. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Mail Stop Reissue, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD
First Named Inventor/Applicant Name:	Anatoli Gorchetchnikov
Filer:	Christopher M. Colice/Lindsey Lovato
Attorney Docket Number:	NRLA-001/03US 320730-2040

Filed as Small Entity

Filing Fees for Reissue (Utility)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
UTILITY REISSUE BASIC	2014	1	140	140
DESIGN AND UTILITY REISSUE BASIC	2114	1	300	300
DESIGN AND UTILITY REISSUE BASIC	2314	1	1080	1080
Pages:				
Claims:				
REISSUE CLAIMS IN EXCESS OF 20 FOR SMALL	2205	116	40	4640
REISSUE- INDEPENDENT CLAIMS	2204	13	210	2730
Miscellaneous-Filing:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**8890**

Electronic Acknowledgement Receipt

EFS ID:	30898818
Application Number:	15808201
International Application Number:	
Confirmation Number:	1077
Title of Invention:	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD
First Named Inventor/Applicant Name:	Anatoli Gorchetchnikov
Customer Number:	58249
Filer:	Christopher M. Colice/Lindsey Lovato
Filer Authorized By:	Christopher M. Colice
Attorney Docket Number:	NRLA-001/03US 320730-2040
Receipt Date:	09-NOV-2017
Filing Date:	
Time Stamp:	15:18:29
Application Type:	Reissue (Utility)

Payment information:

Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	$8890
RAM confirmation Number	111317INTEFSW00001830501283
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Consent of Assignee accompanying the declaration	NRLA00103US-Consent.pdf	347956 9c447d95a0b38f8e75d61f8e9e6680051992e1e1	no	2
Warnings:					
Information:					
2	Specification	NRLA00103US-Specification.pdf	15084626 838da5e1d7b4b54d557d38b577ab18bdacfa214b	no	7
Warnings:					
Information:					
3	Claims	NRLA00103US-Caims.pdf	25084 cffbfe49c986badbd06c076ed97b53e7cbecdac3	no	2
Warnings:					
Information:					
4	Drawings-only black and white line drawings	NRLA00103US-Drawings.pdf	127960 dcf10ebdd344106b5252f8bde044406d5c43af92	no	5
Warnings:					
Information:					
5	Power of Attorney	NRLA00103US-POA.pdf	894017 3450592efd486f762d2affdd4eb27ac356870045	no	2
Warnings:					
Information:					
6	Terminal Disclaimer Filed	NRLA00103US-TD.pdf	63736 42484818f20d37e5051e40b295e787f9c03509f8	no	1
Warnings:					
Information:					

7	Assignee showing of ownership per 37 CFR 3.73	NRLA00103US-STATEMENT.pdf	83905 34be19e014143cc9425a7bef7b64b59b58a3bd69	no	2
Warnings:					
Information:					
8	Application Data Sheet	WebADS.pdf	157039 2aa29746ce80881b5c0064ec57fd4077457500a7	no	9
Warnings:					
Information:					
9	Oath or Declaration filed	NRLA00103US-DECS.pdf	1521158 5529d0a46046e782f83db40e8024a851ebe2ef3c	no	8
Warnings:					
Information:					
10		NRLA00103US-PRELIMAMENDMENT.pdf	201062 a9f5d8814323af4f3cbe4421f145b12f4154af70	yes	24

Multipart Description/PDF files in .zip description

Document Description	Start	End
Applicant Arguments/Remarks Made in an Amendment	23	24
Claims	2	22
Preliminary Amendment	1	1

Warnings:					
Information:					
11	Transmittal Reissue Application	NRLA00103US-TRANSMITTAL2.pdf	303369 cdf0f5399e483631b5b6f16b6f36c8b3a2ad3eef	no	2
Warnings:					
Information:					
12	Fee Worksheet (SB06)	fee-info.pdf	38131 1c8959bed43a0514896f93626a0dc887ab0a7fbe	no	2
Warnings:					

Information:	
Total Files Size (in bytes):	18848043

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/AIA/53 (09-12)
Approved for use through 08/31/2013. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REISSUE APPLICATION: CONSENT OF ASSIGNEE; STATEMENT OF NON-ASSIGNMENT	Docket Number (Optional) NRLA-001/03US

This is part of the application for a reissue patent based on the original patent identified below.

Name of Patentee(s)
Neurala, Inc.

Patent Number	Date Patent Issued
9,189,828	11/17/2015

Title of Invention
GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD

1. ☑ Filed herein is a statement under 37 CFR 3.73(c). (Form PTO/AIA/96)

2. ☐ Ownership of the patent is in the inventor(s), and no assignment of the patent is in effect.

One of boxes 1 or 2 above must be checked. If multiple assignees, complete this form for each assignee. If box 2 is checked, skip the next entry and go directly to "Name of Assignee."

The written consent of all assignees and inventors owning an undivided interest in the original patent is included in this application for reissue.

The assignee(s) owning an undivided interest in said original patent is/are Neurala, Inc., and the assignee(s) consents to the accompanying application for reissue.

Name of assignee/inventor (if not assigned)
Neurala, Inc.

Signature	Date
/Anatoly Gorshechnikov/	11/9/2017

Typed or printed name and title of person signing for assignee (if assigned)

Anatoly Gorshechnikov, CTO

This collection of information is required by 37 CFR 1.172. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 6 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD

RELATED APPLICATIONS

The present application claims a priority benefit, under 35 U.S.C. §120, as a continuation of U.S. application Ser. No. 11/860,254, now U.S. Pat. No. 8,648,867 B2, filed Sep. 24, 2007, entitled "Graphic Processor Based Accelerator System and Method," which claims the priority benefit, under 35 U.S.C. §119(e), of U.S. Application No. 60/826,892, filed Sep. 25, 2006. Each of the above-identified applications is incorporated herein by reference in its entirety.

BACKGROUND

Graphics Processing Units (GPUs) are found in video adapters (graphic cards) of most personal computers (PCs), video game consoles, workstations, etc. and are considered highly parallel processors dedicated to fast computation of graphical content. With the advances of the computer and console gaming industries, the need for efficient manipulation and display of 3D graphics has accelerated the development of GPUs.

In addition, manufacturers of GPUs have included general purpose programmability into the GPU architecture leading to the increased popularity of using GPUs for highly parallelizable and computationally expensive algorithms outside of the computer graphics domain. When implemented on conventional video card architectures, these general purpose GPU (GPGPU) applications are not able to achieve optimal performance, however. There is overhead for graphics-related features and algorithms that are not necessary for these non-video applications.

SUMMARY

Numerical simulations, e.g., finite element analysis, of large systems of similar elements (e.g. neural networks, genetic algorithms, particle systems, mechanical systems) are one example of an application that can benefit from GPGPU computation. During numerical simulations, disk and user input/output can be performed independently of computation because these two processes require interactions with peripheral hardware (disk, screen, keyboard, mouse, etc) and put relatively low load on the central processing unit/system (CPU). Complete independence is not desirable, however; user input might affect how the computation is performed and even interrupt it if necessary. Furthermore, the user output and the disk output are dependent on the results of the computation. A reasonable solution would be to separate input/output into threads, so that it is interacting with hardware occurs in parallel with the computation. In this case whatever CPU processing is required for input/output should be designed so that it provides the synchronization with computation.

In the case of GPGPU, the computation itself is performed outside of the CPU, so the complete system comprises three "peripheral" components: user interactive hardware, disk hardware, and computational hardware. The central processing unit (CPU) establishes communication and synchronization between peripherals. Each of the peripherals is preferably controlled by a dedicated thread that is executed in parallel with minimal interactions and dependencies on the other threads.

A GPU on a conventional video card is usually controlled through OpenGL, DirectX, or similar graphic application programming interfaces (APIs). Such APIs establish the context of graphic operations, within which all calls to the GPU are made. This context only works when initialized within the same thread of execution that uses it. As a result, in a preferred embodiment, the context is initialized within a computational thread. This creates complications, however, in the interaction between the user interface thread that changes parameters of simulations and the computational thread that uses these parameters.

A solution as proposed here is an implementation of the computational stream of execution in hardware, so that thread and context initialization are replaced by hardware initialization. This hardware implementation includes an expansion card comprising a printed circuit board having (a) one or more graphics processing units, (b) two or more associated memory banks that are logically or physically partitioned, (c) a specialized controller, and (d) a local bus providing signal coupling compatible with the PCI industry standards (this includes but is not limited to PCI-Express, PCI-X, USB 2.0, or functionally similar technologies). The controller handles most of the primitive operations needed to set up and control GPU computation. As a result, the CPU is freed from this function and is dedicated to other tasks. In this case a few controls (simulation start and stop signals from the CPU and the simulation completion signal back to CPU), GPU programs and input/output data are the information exchanged between CPU and the expansion card. Moreover, since on every time step of the simulation the results from the previous time step are used but not changed, the results are preferably transferred back to CPU in parallel with the computation.

In general, according to one aspect, the invention features a computer system. This system comprises a central processing unit, main memory accessed by the central processing unit, and a video system for driving a video monitor in response to the central processing unit as is common. The computer system further comprises an accelerator that uses input data from and provides output data to the central processing unit. This accelerator comprises at least one graphics processing unit, accelerator memory for the graphic processing unit, and an accelerator controller that moves the input data into the at least one graphics processing unit and the accelerator memory to generate the output data.

In the preferred, the central processing unit transfers the input data for a simulation to the accelerator, after which the accelerator executes simulation computations to generate the output data, which is transferred to the central processing unit. Preferably, the accelerator controller dictates an order of execution of instructions to the at least one graphics processing unit. The use of the separate controller enables data transfer during execution such that the accelerator controller transfers output data from the accelerator memory to main memory of the central processing unit.

In the preferred embodiment, the accelerator controller comprises an interface controller that enables the accelerator to communicate over a bus of the computer system with the central processing unit.

In general according to another aspect, the invention also features an accelerator system for a computer system, which comprises at least one graphics processing unit, accelerator memory for the graphic processing unit and an accelerator controller for moving data between the at least one graphics processing unit and the accelerator memory.

In general according to another aspect, the invention also features a method for performing numerical simulations in a computer system. This method comprises a central processing unit loading input data into an accelerator system from main memory of the central processing unit and an accelera-

tor controller transferring the input data to a graphics processing unit with instructions to be performed on the input data. The accelerator controller then transfers output data generated by the graphic processing unit to the central processing unit as output data.

The above and other features of the invention including various novel details of construction and combinations of parts, and other advantages, will now be more particularly described with reference to the accompanying drawings and pointed out in the claims. It will be understood that the particular method and device embodying the invention are shown by way of illustration and not as a limitation of the invention. The principles and features of this invention may be employed in various and numerous embodiments without departing from the scope of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

In the accompanying drawings, reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale; emphasis has instead been placed upon illustrating the principles of the invention. Of the drawings:

FIG. **1** is a schematic diagram illustrating a computer system including the GPU accelerator according to an embodiment of the present invention;

FIG. **2** is block diagram illustrating the architecture for the GPU accelerator according to an embodiment of the present invention;

FIG. **3** is a block/flow diagram illustrating an exemplary implementation of the top level control of the GPU accelerator system;

FIG. **4** is a flow diagram illustrating an exemplary implementation of the bottom level control of the GPU accelerator system that is used to execute the target computation; and

FIG. **5** is an example population of nine computational elements arranged in a 3×3 square and a potential packing scheme for texture pixels, according to an implementation of the present invention.

DETAILED DESCRIPTION

FIG. **1** shows a computer system **100** that has been constructed according to the principles of the present invention.

In more detail, the computer system **100** in one example is a standard personal computer (PC). However, this only serves as an example environment as computing environment **100** does not necessarily depend on or require any combination of the components that are illustrated and described herein. In fact, there are many other suitable computing environments for this invention, including, but not limited to, workstations, server computers, supercomputers, notebook computers, hand-held electronic devices such as cell phones, mp3 players, or personal digital assistants (PDAs), multiprocessor systems, programmable consumer electronics, networks of any of the above-mentioned computing devices, and distributed computing environments that including any of the above-mentioned computing devices.

In one implementation the GPU accelerator is implemented as an expansion card **180** includes connections with the motherboard **110**, on which the one or more CPU's **120** are installed along with main, or system memory **130** and mass/non volatile data storage **140**, such as hard drive or redundant array of independent drives (RAID) array, for the computer system **100**. In the current example, the expansion card **180** communicates to the motherboard **110** via a local bus **190**. This local bus **190** could be PCI, PCI Express, PCI-X, or any other functionally similar technology (depending upon the availability on the motherboard **110**). An external version GPU accelerator is also a possible implementation. In this example, the external GPU accelerator is connected to the motherboard **110** through USB-2.0, IEEE 1394 (Firewire), or similar external/peripheral device interface.

The CPU **120** and the system memory **130** on the motherboard **110** and the mass data storage system **140** are preferably independent of the expansion card **180** and only communicate with each other and the expansion card **180** through the system bus **200** located in the motherboard **110**. A system bus **200** in current generations of computers have bandwidths from 3.2 GB/s (Pentium 4 with AGTL+, Athlon XP with EV6) to around 15 GB/s (Xeon Woodcrest with AGTL+, Athlon 64/Opteron with Hypertransport), while the local bus has maximal peak data transfer rates of 4 GB/s (PCI Express 16) or 2 GB/s (PCI-X 2.0). Thus the local bus **190** becomes a bottleneck in the information exchange between the system bus **200** and the expansion card **180**. The design of the expansion card and methods proposed herein minimizes the data transfer through the local bus **190** to reduce the effect of this bottleneck.

The system memory **130** is referred to as the main random-access memory (RAM) in the description herein. However, this is not intended to limit the system memory **130** to only RAM technology. Other possible computer storage media include, but are not limited to ROM, EEPROM, flash memory, or any other memory technology.

In the illustrated example, the GPU accelerator system is implemented on an expansion card **180** on which the one or more GPU's **240** are mounted. It should be noted that the GPU accelerator system GPU **240** is separate from and independent of any GPU on the standard video card **150** or other video driving hardware such as integrated graphics systems. Thus the computations performed on the expansion card **180** do not interfere with graphics display (including but not limited to manipulation and rendering of images).

Various brand of GPU are relevant. Under current technology, GPU's based on the GeForce series from NVIDIA Corporation or the Catalyst series from ATI/Advanced Micro Devices, Inc.

The output to a video monitor **170** is preferably through the video card **150** and not the GPU accelerator system **180**. The video card **150** is dedicated to the transfer of graphical information and connects to the motherboard **110** through a local bus **160** that is sometimes physically separate from the local bus **190** that connects the expansion card **180** to the motherboard **110**.

FIG. **2** is a block diagram illustrating the general architecture of the GPU accelerator system and specifically the expansion card **180** in which at least one GPU **240** and associated memories **210** and **250** are mounted. Electrical (signal) and mechanical coupling with a local bus **190** provides signal coupling compatible with the PCI industry standards (this includes but is not limited to PCI, PCI-X, PCI Express, or functionally similar technology).

The GPU accelerator further preferably comprises one specifically designed accelerator controller **220**. Depending upon the implementation, the accelerator controller **220** is field programmable gate array (FPGA) logic, or custom built application-specific (ASIC) chip mounted in the expansion card **180**, and in mechanical and signal coupling with the GPU **240** and the associated memories **210** and **250**. During initial design, a controller can be partially or even fully implemented in software, in one example.

The controller **220** commands the storage and retrieval of arrays of data (on a conventional video card the arrays of data are represented as textures, hence the term 'texture' in this document refers to a data array unless specified otherwise and each element of the texture is a pixel of color information), execution of GPU programs (on a conventional video card these programs are called shaders, hence the term 'shader' in this document refers to a GPU program unless specified otherwise), and data transfer between the system bus **200** and the expansion card **180** through the local bus **190** which allows communication between the main CPU **120**, RAM **130**, and disk **140**.

Two memory banks **210** and **250** are mounted on the expansion card **180**. In some example, these memory banks separated in the hardware, as shown, or alternatively implemented as a single, logically partitioned memory component.

The reason to separate the memory into two partitions **210** **250** stems from the nature of the computations to which the GPU accelerator system is applied. The elements of computation (computational elements) are characterized by a single output variable. Such computational elements often include one or more equations. Computational elements are same or similar within a large population and are computed in parallel. An example of such a population is a layer of neurons in an artificial neural network (ANN), where all neurons are described by the same equation. As a result, some data and most of the algorithms are common to all computational elements within population, while most of the data and some algorithms are specific for each equation. Thus, one memory, the shader memory bank **210**, is used to store the shaders needed for the execution of the required computations and the parameters that are common for all computational elements and is coupled with the controller **220** only. The second memory, the texture memory bank **250**, is used to store all the necessary data that are specific for every computational element (including, but not limited to, input data, output data, intermediate results, and parameters) and is coupled with both the controller **220** and the GPU **240**.

The texture memory bank **250** is preferably further partitioned into four sections. The first partition **250** a is designed to hold the external input data patterns. The second partition **250***b* is designed to hold the data textures representing internal variables. The third partition **250***c* is designed to hold the data textures used as input at a particular computation step on the GPU **240**. The fourth partition **250***d* holds the data textures used to accommodate the output of a particular computational step on the GPU **240**. This partitioning scheme can be done logically, does not require hardware implementation. Also the partitioning scheme is also altered based on new designs or needs of the algorithms being employed. The reason for this partitioning is further explained in the Data Organization section, below.

A local bus interface **230** on the controller **220** serves as a driver that allows the controller **220** to communicate through the local bus **190** with the system bus **200** and thus the CPU **120** and RAM **130**. This local bus interface **230** is not intended to be limited to PCI related technology. Other drivers can be used to interface with comparable technology as a local bus **190**.

Data Organization

Each computational element discussed above has output variables that affect the rest of the system. For example in the case of a neural network it is the output of a neuron. A computational element also usually has several internal variables that are used to compute output variables, but are not exposed to the rest of the system, not even to other elements of the same population, typically. Each of these variables is represented as a texture. The important difference between output variables and internal variables is their access.

Output variables are usually accessed by any element in the system during every time step. The value of the output variable that is accessed by other elements of the system corresponds to the value computed on the previous, not the current, time step. This is realized by dedicating two textures to output variables—one holds the value computed during the previous time step and is accessible to all computational elements during the current time step, another is not accessible to other elements and is used to accumulate new values for the variable computed during the current time step. In-between time steps these two textures are switched, so that newly accumulated values serve as accessible input during the next time step, while the old input is replaced with new values of the variable. This switch is implemented by swapping the address pointers to respective textures as described in the System and Framework section.

Internal variables are computed and used within the same computational element. There is no chance of a race condition in which the value is used before it is computed or after it has already changed on the next time step because within an element the processing is sequential. Therefore, it is possible to render the new value of internal variable into the same texture where the old was read from in the texture memory bank. Rendering to more than one texture from a single shader is not implemented in current GPU architectures, so computational elements that track internal variables would have to have one shader per variable. These shaders can be executed in order with internal variables computed first, followed by output variables.

Further savings of texture memory is achieved through using multiple color components per pixel (texture element) to hold data. Textures can have up to four color components that are all processed in parallel on a GPU. Thus, to maximize the use of GPU architecture it is desirable to pack the data in such a way that all four components are used by the algorithm. Even though each computational element can have multiple variables, designating one texture pixel per element is ineffective because internal variables require one texture and output variables require two textures. Furthermore, different element types have different numbers of variables and unless this number is precisely a multiple of four, texture memory can be wasted.

A more reasonable packing scheme would be to pack four computational elements into a pixel and have separate textures for every variable associated with each computational element. In this case the packing scheme is identical for all textures, and therefore can be accessed using the same algorithm. Several ways to approach this packing scheme are outlined here. An example population of nine computational elements arranged in a 3×3 square (FIG. **5***a*) can be packed by element (FIG. **5***b*), by row (FIG. **5***c*), or by square (FIG. **5***d*).

Packing by element (FIG. **5***b*) means that elements **1**,**2**,**3**,**4** go into first pixel; **5**,**6**,**7**,**8** go into second pixel; **9** goes into third pixel. This is the most compact scheme, but not convenient because the geometrical relationship is not preserved during packing and its extraction depends on the size of the population.

Packing by row (column; FIG. **5***c*) means that elements **1**,**2**,**3** go into pixel (**1**,**1**); **3**,**4**,**5** go into pixel (**2**,**1**), **7**,**8**,**9** go into pixel (**3**,**1**). With this scheme the element's y coordinate in the population is the pixel's y coordinate, while the element's x coordinate in the population is the pixel's x coordinate times four plus the index of color component. Five by five populations in this case will use 2×5 texture, or 10 pixels. Five of these pixels will only use one out of four components, so it

wastes 37.5% of this texture. 25×1 population will use 6×1 texture (six pixels) and will waste 12.5% of it.

Packing by square (FIG. **5***d*) means that elements **1**,**2**,**4**,**5** go into pixel (**1**,**1**); **3**,**6** go into pixel (**1**,**2**); **7**,**8** go into pixel (**2**,**1**), and **9** goes into pixel (**2**,**2**). Both the row and the column of the element are determined from the row (column) of the pixel times two plus the second (first) bit of the color component index. Five by five populations in this case will use 3×3 texture, or 9 pixels. Four of these pixels will only use two out of four components, and one will only use one component, so it wastes 34.4% of this texture. This is more advantageous than packing by row, since the texture is smaller and the waste is also lower. 25×1 population on the other hand will use 13×1 texture (thirteen pixels) and waste >50% of it, which is much worse than packing by row.

In order to eliminate waste altogether the population should have even dimensions in the square packing, and it should have a number of columns divisible by four in row packing. Theoretically, the chances are approximately equivalent for both of these cases to occur, so the particular task and data sizes should determine which packing scheme is preferable in each individual case.

The System and Framework

FIG. **3** shows an exemplary implementation of the top level system and method that is used to control the computation. It is a representation of one of several ways in which a system and method for processing numerical techniques can be implemented in the invention described herein and so the implementation is not intended to be limited to the following description and accompanying figure.

The method presented herein includes two execution streams that run on the CPU **120**—User Interaction Stream **302** and Data Output Stream **301**. These two streams preferably do not interact directly, but depend on the same data accumulated during simulations. They can be implemented as separate threads with shared memory access and executed on different CPUs in the case of multi-CPU computing environment. The third execution stream—Computational Stream **303**—runs on the GPU accelerator of the expansion card **180** and interacts with the User Interaction Stream **302** through initialization routines and data exchange in between simulations. The Computational Stream **303** interacts with the User Interaction Stream and the Data Output Stream through synchronization procedures during simulations.

The crucial feature of the interaction between the User Interaction Stream **302** and the Computational Stream **303** is the shift of priorities. Outside of the simulation, the system **100** is driven by the user input, thus the User Interaction Stream **302** has the priority and controls the data exchange **304** between streams. After the user starts the simulation, the Computational Stream **303** takes the priority and controls the data exchange between streams until the simulation is finished or interrupted **350**.

The user starts **300** the framework through the means of an operating system and interacts with the software through the user interaction section **305** of the graphic user interface **306** executed on the CPU **120**. The start **300** of the implementation begins with a user action that causes a GUI initialization **307**, Disk input/output initialization **308** on the CPU **120**, and controller initialization **320** of the GPU accelerator on the expansion card **180**. GUI initialization includes opening of the main application window and setting the interface tools that allow the user to control the framework. Disk I/O initialization can be performed at the start of the framework, or at the start of each individual simulation.

The user interaction **305** controls the setting and editing of the computational elements, parameters, and sources of external inputs. It specifies which equations should have their output saved to disk and/or displayed on the screen. It allows the user to start and stop the simulation. And it performs standard interface functions such as file loading and saving, interactive help, general preferences and others.

The user interaction **305** directs the CPU **120** to acquire the new external input textures needed (this includes but is not limited to loading from disk **140** or receiving them in real time from a recording device), parses them if necessary **309**, and initializes their transfer to the expansion card **180**, where they are stored **325** in the texture memory bank **250** by the controller **220**. The user interaction **305** also directs the CPU **120** to parse populations of elements that will be used in the simulation, convert them to GPU programs (shaders), compile them **310**, and initializes their transfer to the expansion card **180**, where they are stored **326** in the shader memory bank **210** by the controller **220**. This operation is accompanied by the upload **309** of the initial data into the input partition of the texture memory bank **250**, and stores the shader order of execution in the controller **220**. The user can perform operations **309** and **310** as many times as necessary prior to starting the simulation or between simulations.

The editing of the system between simulations is difficult to accomplish without the hardware implementation of the computational thread suggested herein. The system of equations (computational elements) is represented by textures that track variables plus shaders that define processing algorithms. As mentioned above, textures, shaders and other graphics related constructs can only be initialized within the rendering context, which is thread specific. Therefore textures and shaders can only be initialized in the computational thread.

Network editing is a user-interactive process, which according to the scheme suggested above happens in the User Interaction Stream **302**. The simulation software thus has to take the new parameters from the User Interaction Stream **302**, communicate them to the Computational Stream **303** and regenerate the necessary shaders and textures. This is hard to accomplish without a hardware implementation of the Computational Stream **303**. The Computational Stream **303** is forked from the User Interaction Stream and it can access the memory of the parent thread, but the reverse communication is harder to achieve. The controller **220** allows operations **309** and **310** to be performed as many times as necessary by providing the necessary communication to the User Interaction Stream **302**.

After execution of the input parser texture generation **309** and population parser shader generator and compiler **310** are performed at least once, the user has the option to initialize the simulation **311**. During this initialization the main control of the framework is transferred to the GPU accelerator system's accelerator controller **220** and computation **330** is started (see FIG. **4**; **420**). The user retains the ability to interrupt the simulation, change the input, or to change the display properties of the framework, but these interactions are queued to be performed at times determined by the controller-driven data exchange **314** and **316** to avoid the corruption of the data.

The progress monitor **312** is not necessary for performance, but adds convenience. It displays the percentage of completed time steps of the simulation and allows the user to plan the schedule using the estimates of the simulation wall clock times. Controller-driven data exchange **314** updates the display of the results **313**. Online screen output for the user selected population allows the user to monitor the activity and evaluate the qualitative behavior of the network. Simulations with unsatisfactory behavior can be terminated early to change parameters and restart. Controller-driven data

exchange **314** also drives the output of the results to disk **317**. Data output to disk for convenience can be done on an element per file basis. A suggested file format includes a leftmost column that displays a simulated time for each of the simulation steps and subsequent columns that display variable values during this time step in all elements with identical equations (e.g. all neurons in a layer of a neural network).

Controller-driven data exchange or input parser texture generator **316** allows the user to change input that is generated on the fly during the simulation. This allows the framework monitoring of the input that is coming from a recording device (video camera, microphone, cell recording electrode, etc) in real time. Similar to the initial input parser **309**, it preprocesses the input into a universal format of the data array suitable for texture generation and generates textures. Unlike the initial parser **309**, here the textures are transferred to hardware not whenever ready but upon the request of the controller **220**.

The controller **220** also drives the conditional testing **315** and **318** informs the CPU-bound streams whether the simulation is finished. If so, the control returns to the User Interaction Stream. The user then can change parameters or inputs (**309** and **310**), restart the simulation (**311**) or quit the framework (**390**).

SANNDRA (Synchronous Artificial Neuronal Network Distributed Runtime Algorithm; http://www.kimess.net/Docs/SANNDRA/html) was developed to accelerate and optimize processing of numerical integration of large non-homogenous systems of differential equations. This library is fully reworked in its version 2.x.x to support multiple computational backends including those based on multicore CPUs, GPUs and other processing systems. GPU based backend for SANNDRA-2.x.x can serve as an example practical software implementation of the method and architecture described above and pictorially represented in FIG. **3**.

To use SANNDRA, the application should create a TSimulator object either directly or through inheritance. This object will handle global simulation properties and control the User Interaction Stream, Data Output Stream, and Computational Stream. Through TSimulator::timestep( ), TSimulator::outfileInterval( ), and TSimulator::outmode( ), the application can set the time step of the simulation, the time step of disk output, and the mode of the disk output. The external input pattern should be packed into a TPattern object and bound to the simulation object through TSimulator::resetInputs( ) method. TSimulator::simLength( ) sets the length of the simulation.

The second step is to create at least one population of equations (Tpopulation object). Population holds one equation object TEquation. This object contains only a formula and does not hold element-specific data, so all elements of the population can share single TEquation.

The TEquation object is converted to a GPU program before execution. GPU programs have to be executed within a graphical context, which is stream specific. TSimulator creates this context within a Computational Stream, therefore all programs and data arrays that are necessary for computation have to be initialized within Computational Stream. Constructor of TPopulation is called from User Interaction Stream, so no GPU-related objects can be initialized in this constructor.

TPopulation::fillElements( ) is a virtual method designed to overcome this difficulty. It is called from within the Computational Stream after TSimulator::networkCreate( ) is called in the User Interaction Stream. A user has to override TPopulation::fillElements( ) to create TEquation and other computation related objects both element independent and element-specific. Element independent objects include subcomponents of TEquation and objects that describe how to handle interdependencies between variables implemented through derivatives of TGate class.

Element-specific data is held in TElement objects. These objects hold references to TEquation and a set of TGate objects. There is one TElement per population, but the size of data arrays within this object corresponds to population size. All TElement objects have to be added to the TSimulator list of elements by calling TSimulator::addUnit( ) method from TPopulation::fillElements( ).

Finally, TPopulation::fillElements( ) should contain a set of TElement::add*Dependency( ) calls for each element. Each of these calls sets a corresponding dependency for every TGate object. Here TGate object holds element independent part of dependency and TElement::add*Dependency( ) sets element-specific details.

System provided TPopulation handles the output of computational elements, both when they need to exchange the data and when they need to output it to disk. User implementation of TPopulation derivative can add screen output.

Listing 1 is an example code of the user program that uses a recurrent competitive field (RCF) equation:

Listing 1

```
uint16_t w = 3, h = 3;
static float m_compet = 0.5;
static float m_persist = 1.0;
class TCablePopRCF : public TPopulation
{
TEq_RCF* m_equation;
TGate* m_gate1;
TGate* m_gate2;
void createGatingStructure( )
{
m_gate1 = new TGate(0);
m_gate2 = new TGate(1);
};
void createUnitStructure(TBasicUnit* u)
{
u->addO2OPInputDependency(m_gate1, 0., 0., 0.004, 0., 0, 0);
u->addFullDependency(m_gate2, population( ));
}
public: TCablePopRCF( ) : TPopulation("compCPU RCF", w, h, true) { };
~TCablePopRCF( ) {if(m_equation) delete m_equation;
   if(m_gate1) delete m_gate1;
   if(m_gate2) delete m_gate2;};
bool fillElements(TSimulator* sim);
};
bool TCablePopRCF::fillElements(TSimulator* sim)
{
m_equation = new TEq_RCF(this, m_compet, m_persist);
createGatingStructure( );
for(size_t i = 0; i < xSize( ); ++i)
for(size_t j = 0; j < ySize( ); ++j)
{
TElement* u = new TCPUElement(this, m_equation, i, j);
sim->addUnit(u);
createUnitStructure(u);
}
Return true;
}
int
main( )
{
// Input pattern generation (309 in FIG.3)
uint32_t* pat = new uint32_t[w*h];
TRandom<float> randGen (0);
for(uint32_t I = 0; I < w*h; ++i)
pat[i] = randGen.random( );
Tpattern* p = new Tpattern(pat, w, h);
// Setting up the simulation
TSimulator* cableSim = new TSimulator("data"); //(308 and 320 in
FIG. 3)
```

-continued

Listing 1

```
cableSim->timestep(0.05); //(320 in FIG. 3)
cableSim->resetInputs(p); //(325 in FIG. 3)
cableSim->outfileInterval(0.1); //(308 in FIG. 3)
cableSim->outmode(SANNDRA::timefunc); //(308 in FIG. 3)
cableSim->simLength(60.0); //(320 in FIG. 3)
// Preparing the population
TPopulation* cablePop = new TCablePopRCF( ); //(310 in FIG. 3)
cableSim->networkCreate( ); //(326 in FIG. 3)
uint16_t user= 1;
while(user)
{
if(! cableSim->simulationStart(true, 1)) //(311 in FIG. 3)
exit(1);
std::cout<<"Repeat?\n"; //(305 in FIG. 3)
std::cin>>user; //(305 in FIG. 3)
if(user == 1)
cableSim->networkReset( ); //(305 in FIG. 3)
{
If(cableSim)
Delete cableSim; //Also deletes cablePop and its internals
exit(0);
};
```

FIG. **4** is a detailed flow diagram illustrating a part of an exemplary implementation of the bottom level system and method performed during the computation on the GPU accelerator of the expansion card **180** and is a more detailed view of the computational box **330** in FIG. **3**. FIG. **4** is a representation of one of several ways in which a system and method for processing numerical techniques can be implemented.

With systems of equations that have complex interdependencies it is likely that the variable in some equation from a previous time step has to be used by some other equation after the new values of this variable are already computed for new time step. To avoid data confusion, the new values of variables should be rendered in a separate texture. After the time step is completed for all equations, these new values should be copied over old values so that they are used as input during the next time step. Copying textures is an expensive operation, computationally, but since the textures are referred to by texture IDs (pointers), swapping these pointers for input and output textures after each time step achieves the same result at a much lesser cost.

In the hardware solution suggested herein, ID swapping is equivalent to swapping the base memory address for two partitions of the texture memory bank **250**. They are swapped **485** during synchronization (**485**, **430**, and **455**) so that data transfer **445** and the computation **435**-**487** proceeds immediately and in parallel with data transfer as shown in FIG. **4**. A hardware solution allows this parallelism through access of the controller **220** to the onboard texture memory bank **250**.

The main computation and data exchange are executed by the controller **220**. It runs three parallel substreams of execution: Computational Substream **403**, Data Output Substream **402**, and Data Input Substream **404**. These streams are synchronized with each other during the swap of pointers **485** to the input and output texture memory partitions of the texture memory bank **250** and the check for the last iteration **487**. Algorithmically, these two operations are a single atomic operation, but the block diagram shows them as two separate blocks for clarity.

The Computational Substream **403** performs a computational cycle including a sequential execution of all shaders that were stored in the shader memory bank **210** using the appropriate input and output textures. To begin the simulation the controller **220** initializes three execution substreams **403**, **402**, and **404**. On every simulation step, the Computational Substream **403** determines which textures the GPU **240** will need to perform the computations and initiates the upload **435** of them onto the GPU **240**. The GPU **240** can communicate directly with the texture memory bank **250** to upload the appropriate texture to perform the computations. The controller **220** also pulls the first shader (known by the stored order) from the shader memory bank **210** and uploads **450** it onto the GPU **240**.

The GPU **240** executes the following operations in this order: performs the computation (execution of the shader) **470**; tells the controller **220** that it is done with the computations for the current shader; and after all shaders for this particular equation are executed sends **480** the output textures to the output portion of the texture memory bank **250**. This cycle continues through all of the equations based on the branching step **482**.

An example shader that performs fourth order Runge-Kutta numerical integration is shown in Listing 2 using GLSL notation;

Listing 2

```
uniform sampler2DRect Variable;
uniform float integration_step;
float halfstep = integration_step*0.5;
float fl_6step = integration_step/6.0;
vec4 output = texture2DRect(Variable, gl_TexCoord[0].st);
// define equation( ) here
vec4 rungekutta4(vec4 x)
{
const vec4 k1 = equation(x);
const vec4 k2 = equation(x + halfstep*k1);
const vec4 k3 = equation(x + halfstep*k2);
const vec4 k4 = equation(x + integration step*k3);
return fl_6step*(k1 + 2.0*(k2 + k3) + k4);
}
Void main(void)
{
output += rungekutta4(output);
gl_FragColor = output;
}
```

The shader in Listing 2 can be executed on conventional video card. Using the controller **220** this code can be further optimized, however. Since the integration step does not change during the simulation, the step itself as well as the halfstep and ⅙ of the step can be computed once per simulation, and updated in all shaders by a shader update procedures **310**, **326** discussed above.

After all of the equations in the computational cycle are computed the main execution substream **403** on the controller **220** can switch **485** the reference pointers of the input and output portions of the texture memory bank **250**.

The two other substreams of execution on the controller **220** are waiting (blocks **430** and **455**, respectively) for this switch to begin their execution. The Data Input Substream **404** is controlling **440** the input of additional data from the CPU **120**. This is necessary in cases where the simulation is monitoring the changing input, for example input from a video camera or other recording device in the real time. This substream uploads new external input from the CPU **120** to the texture memory bank **250** so it can be used by the main computational substream **403** on the next computational step and waits for the next iteration **475**. The Data Output Substream **445** controls the output of simulation results to the CPU **120** if requested by the user. This substream uploads the results of the previous step to the main RAM **130** so that the CPU **120** can save them on disk **140** or show them on the results display **313** and waits for the next iteration **460**.

Since the Computational Substream **403** determines the timing of input **440** and output **445** data transfers, these data transfers are driven by the controller **220**. To further reduce the data transfer overhead (and disk **140** overhead also) the controller **220** initiates transfer only after selected computational steps. For example, if the experimental data that is simulated was recorded every 10 milliseconds (msec) and the simulation for better precision was computed every 1 msec, then only every tenth result has to be transferred to match the experimental frequency.

This solution stores two copies of output data, one in the expansion card texture memory bank **250** and another in the system RAM **130**. The copy in the system RAM **130** is accessed twice: for disk I/O and screen visualization **313**. An alternative solution would be to provide CPU **120** with a direct read access to the onboard texture memory bank **250** by mapping the memory of the hardware onto a global memory space. The alternative solution will double the communication through the local bus **190**. Since the goal discussed herein is reducing the information transfer through the local bus **190**, the former solution is favored.

The main substream **403** determines if this is the last iteration **487**. If it is the last iteration, the controller **220** waits for the all of the execution substreams to finish **490** and then returns the control to the CPU **120**, otherwise it begins the next computational cycle.

This repeats through all of the computational cycles of the simulation.

CONCLUSION

This GPU accelerator system offers the following potential advantages:

1. Limited computations on the CPU **120**. The CPU **120** is only used for user input, sending information to the controller **220**, receiving output after each computational cycle (or less frequently as defined by the user), writing this output to disk **140**, and displaying this output on the monitor **170**. This frees the CPU **120** to execute other applications and allows the expansion card to run at its full capacity without being slowed down by extensive interactions with the CPU **120**.

2. Minimizing data transfer between the expansion card **180** and the system bus **200**. All of the information needed to perform the simulations will be stored on the expansion card **180** and all simulations will take place on it. Furthermore, whatever data transfer remains necessary will take place in parallel with the computation, thus reducing the impact of this transfer on the performance.

3. New way to execute GPU programs (shaders). Previously, the CPU **120** had full control over the order of shader's execution and was required to produce specific commands on every cycle to tell the GPU **240** which shader to use. With the invention disclosed herein, shaders will initially be stored on the shader memory bank **210** on the expansion card **180** and will be sent to the GPU **240** for execution by the general purpose controller **220** located on the expansion card.

4. Multiple parallelisms. The GPU **240** is inherently parallel and is well suited to perform parallel computations. In parallel with the GPU **240** performing the next calculation, the controller **220** is uploading the data from the previous calculation into main memory **130**. Furthermore, the CPU **120** at the same time uses uploaded previous results to save them onto disk **140** and to display them on the screen through the system bus **200**.

5. Reuse of existing and affordable technology. All hardware used in the invention and mentioned here-in are based on currently available and reliable components. Further advance of these components will provide straightforward improvements of the invention.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

What is claimed is:

1. A computer system, comprising:

a central processing unit to receive input data;

main memory, operably coupled to the central processing unit via a bus, to store the input data received by the central processing unit;

an accelerator, operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory, the accelerator comprising:

at least one graphics processing unit to perform a sequence of computations on the at least a portion of the input data so as to generate output data, intermediate computations in the sequence of computations yielding intermediate results; and

accelerator memory, operably coupled to the graphic processing unit, to store the results of the plurality of sequential computations; and

a controller, operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory, and to transfer at least a portion of the output data from the accelerator memory to the main memory during performance of the sequence of computations by the at least one graphic processing unit.

2. The computer system of claim **1**, wherein the central processing unit is configured to receive the input data in response to a user interaction.

3. The computer system of claim **1**, wherein:

the central processing unit is configured to receive the input data at a first rate; and

the at least one graphics processing unit is configured to perform the sequence of computations at a second rate different than the first rate.

4. The computer system of claim **1**, wherein the main memory is configured to store a copy of the output data stored in the accelerator memory.

5. The computer system of claim **1**, wherein an output of at least one computation in the sequence of computations represents an output of at least one neuron in an artificial neural network.

6. The computer system of claim **1**, wherein accelerator memory comprises:

a first memory bank to store parameters common to all of the computations in the sequence of computations; and

a second memory bank to store data specific to at least one computation in the sequence of computations.

7. The computer system of claim **1**, wherein the controller is configured to transfer the output data from the accelerator memory to the main memory without transferring any of the intermediate results from the accelerator memory to the main memory so as to reduce data transfer via the bus.

8. The computer system of claim **1**, wherein the controller is configured to transfer at least a portion of the output data from the accelerator memory to the main memory after the at least one graphics processing unit has begun to perform another sequence of computations.

Since the Computational Substream **403** determines the timing of input **440** and output **445** data transfers, these data transfers are driven by the controller **220**. To further reduce the data transfer overhead (and disk **140** overhead also) the controller **220** initiates transfer only after selected computational steps. For example, if the experimental data that is simulated was recorded every 10 milliseconds (msec) and the simulation for better precision was computed every 1 msec, then only every tenth result has to be transferred to match the experimental frequency.

This solution stores two copies of output data, one in the expansion card texture memory bank **250** and another in the system RAM **130**. The copy in the system RAM **130** is accessed twice: for disk I/O and screen visualization **313**. An alternative solution would be to provide CPU **120** with a direct read access to the onboard texture memory bank **250** by mapping the memory of the hardware onto a global memory space. The alternative solution will double the communication through the local bus **190**. Since the goal discussed herein is reducing the information transfer through the local bus **190**, the former solution is favored.

The main substream **403** determines if this is the last iteration **487**. If it is the last iteration, the controller **220** waits for the all of the execution substreams to finish **490** and then returns the control to the CPU **120**, otherwise it begins the next computational cycle.

This repeats through all of the computational cycles of the simulation.

CONCLUSION

This GPU accelerator system offers the following potential advantages:

1. Limited computations on the CPU **120**. The CPU **120** is only used for user input, sending information to the controller **220**, receiving output after each computational cycle (or less frequently as defined by the user), writing this output to disk **140**, and displaying this output on the monitor **170**. This frees the CPU **120** to execute other applications and allows the expansion card to run at its full capacity without being slowed down by extensive interactions with the CPU **120**.

2. Minimizing data transfer between the expansion card **180** and the system bus **200**. All of the information needed to perform the simulations will be stored on the expansion card **180** and all simulations will take place on it. Furthermore, whatever data transfer remains necessary will take place in parallel with the computation, thus reducing the impact of this transfer on the performance.

3. New way to execute GPU programs (shaders). Previously, the CPU **120** had full control over the order of shader's execution and was required to produce specific commands on every cycle to tell the GPU **240** which shader to use. With the invention disclosed herein, shaders will initially be stored on the shader memory bank **210** on the expansion card **180** and will be sent to the GPU **240** for execution by the general purpose controller **220** located on the expansion card.

4. Multiple parallelisms. The GPU **240** is inherently parallel and is well suited to perform parallel computations. In parallel with the GPU **240** performing the next calculation, the controller **220** is uploading the data from the previous calculation into main memory **130**. Furthermore, the CPU **120** at the same time uses uploaded previous results to save them onto disk **140** and to display them on the screen through the system bus **200**.

5. Reuse of existing and affordable technology. All hardware used in the invention and mentioned here-in are based on currently available and reliable components. Further advance of these components will provide straightforward improvements of the invention.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.

What is claimed is:

1. A computer system, comprising:
- a central processing unit to receive input data;
- main memory, operably coupled to the central processing unit via a bus, to store the input data received by the central processing unit;
- an accelerator, operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory, the accelerator comprising:
  - at least one graphics processing unit to perform a sequence of computations on the at least a portion of the input data so as to generate output data, intermediate computations in the sequence of computations yielding intermediate results; and
  - accelerator memory, operably coupled to the graphic processing unit, to store the results of the plurality of sequential computations; and
- a controller, operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory, and to transfer at least a portion of the output data from the accelerator memory to the main memory during performance of the sequence of computations by the at least one graphic processing unit.

2. The computer system of claim **1**, wherein the central processing unit is configured to receive the input data in response to a user interaction.

3. The computer system of claim **1**, wherein:
- the central processing unit is configured to receive the input data at a first rate; and
- the at least one graphics processing unit is configured to perform the sequence of computations at a second rate different than the first rate.

4. The computer system of claim **1**, wherein the main memory is configured to store a copy of the output data stored in the accelerator memory.

5. The computer system of claim **1**, wherein an output of at least one computation in the sequence of computations represents an output of at least one neuron in an artificial neural network.

6. The computer system of claim **1**, wherein accelerator memory comprises:
- a first memory bank to store parameters common to all of the computations in the sequence of computations; and
- a second memory bank to store data specific to at least one computation in the sequence of computations.

7. The computer system of claim **1**, wherein the controller is configured to transfer the output data from the accelerator memory to the main memory without transferring any of the intermediate results from the accelerator memory to the main memory so as to reduce data transfer via the bus.

8. The computer system of claim **1**, wherein the controller is configured to transfer at least a portion of the output data from the accelerator memory to the main memory after the at least one graphics processing unit has begun to perform another sequence of computations.

9. The computer system of claim 8, wherein the controller is configured to initiate transfer of the at least a portion of the input data and to transfer the at least a portion of the output data in parallel with performance of at least one computation in the other sequence of computations by the at least one graphics processing unit.

10. The computer system of claim 1, wherein the controller is configured to control execution of the sequence of computations by the at least one graphics processing unit.

11. The computer system of claim 1, further comprising:

at least one of a video camera, a microphone, or a cell recording electrode, operably coupled to the central processor unit, to acquire the input data in real time.

12. A method of performing a sequence of computations on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory, the method comprising:

(A) performing, by the GPU, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data, intermediate computations in the sequence of computations yielding intermediate results;

(B) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the input data from the main memory to the accelerator via the bus; and

(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus.

13. The method of claim 12, further comprising:

storing the input data in the main memory in response to a user interaction.

14. The method of claim 12, further comprising:

receiving the input data at a first rate; and

wherein (A) comprises performing the sequence of computations at a second rate different than the first rate.

15. The method of claim 12, wherein (A) comprises:

generating an output representative of an output of at least one neuron in an artificial neural network.

16. The method of claim 12, wherein (C) comprises:

transferring the second portion of the output data from the accelerator memory to the main memory without transferring any of the intermediate results of the plurality of sequential computations from the accelerator memory to the main memory so as to reduce data transfer via the bus.

17. The method of claim 12, wherein (C) comprises:

transferring the second portion of the output data from the accelerator memory to the main memory after the GPU has begun to perform another sequence of computations.

18. The method of claim 17, wherein (C) further comprises:

initiating transfer of the second portion of the output data in parallel with performance of at least one computation in the other sequence of computations.

19. The method of claim 12, further comprising:

acquiring the input data in real time with at least one of a video camera, a microphone, or a cell recording electrode operably coupled to the CPU.

20. The method of claim 12, further comprising:

storing parameters common to all of the computations in the sequence of computations in a first memory bank in the accelerator memory; and

storing data specific to at least one computation in the sequence of computations in a second memory bank in the accelerator memory.

* * * * *

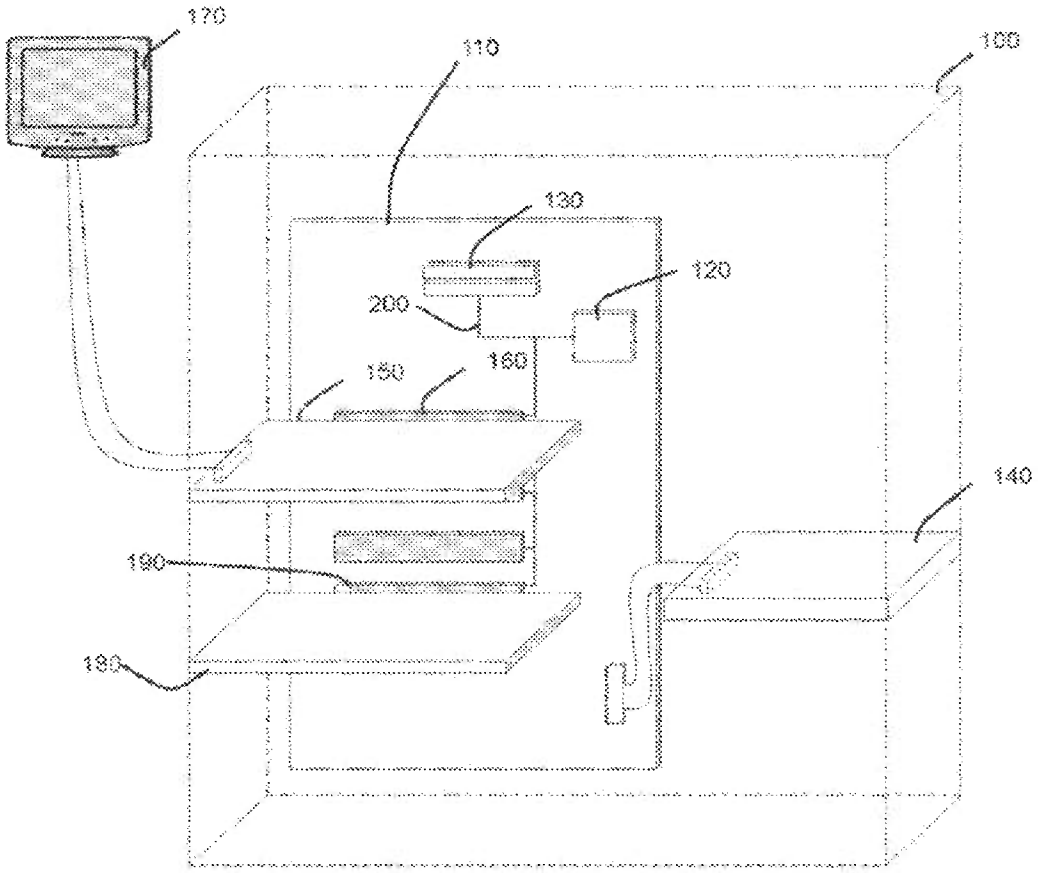


FIG. 1

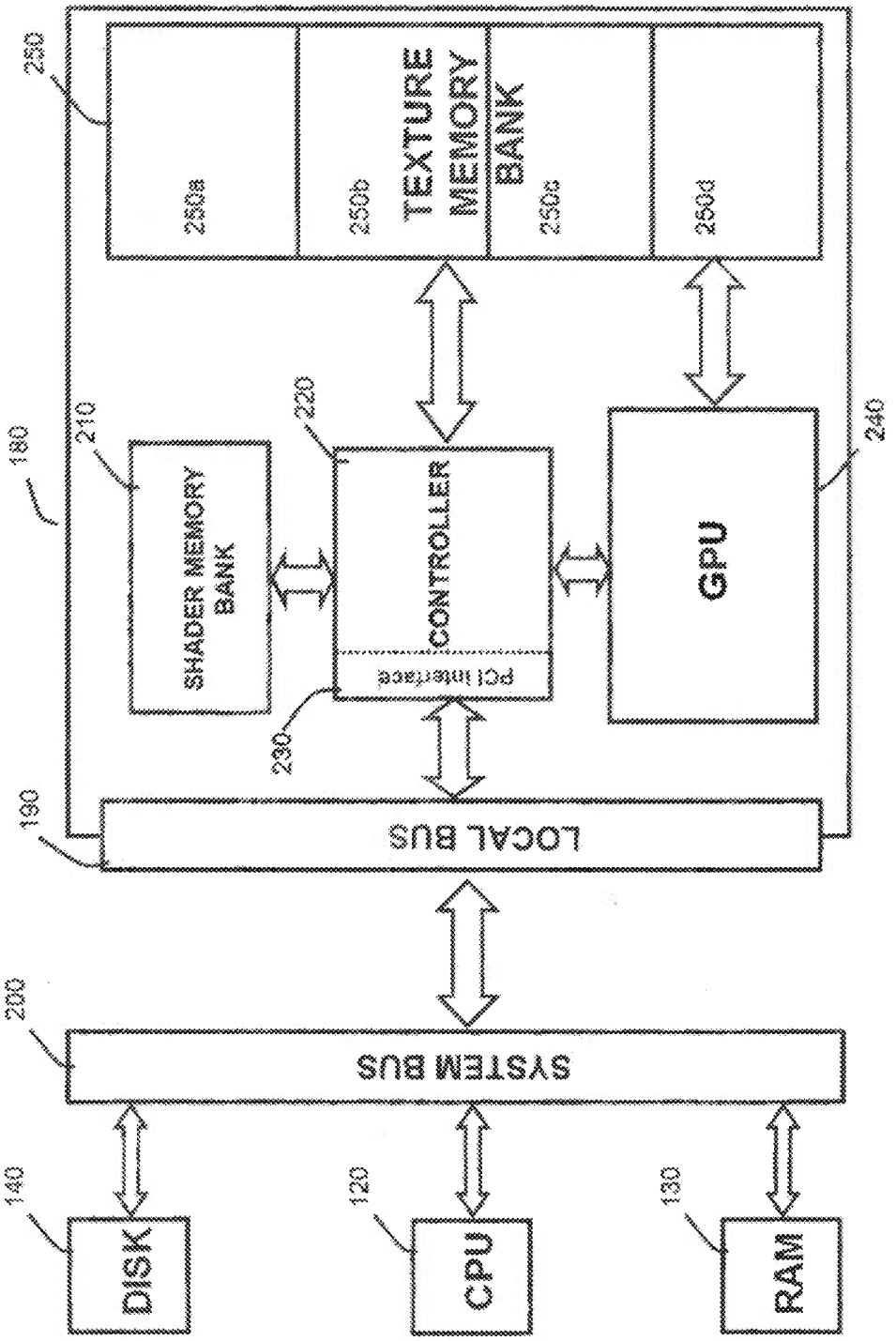

250
250a
250b
TEXTURE MEMORY BANK
250c
250d
180
210
SHADER MEMORY BANK
220
CONTROLLER
PCI Interface
230
240
GPU
190
LOCAL BUS
200
SYSTEM BUS
140
DISK
120
CPU
130
RAM

FIG. 2

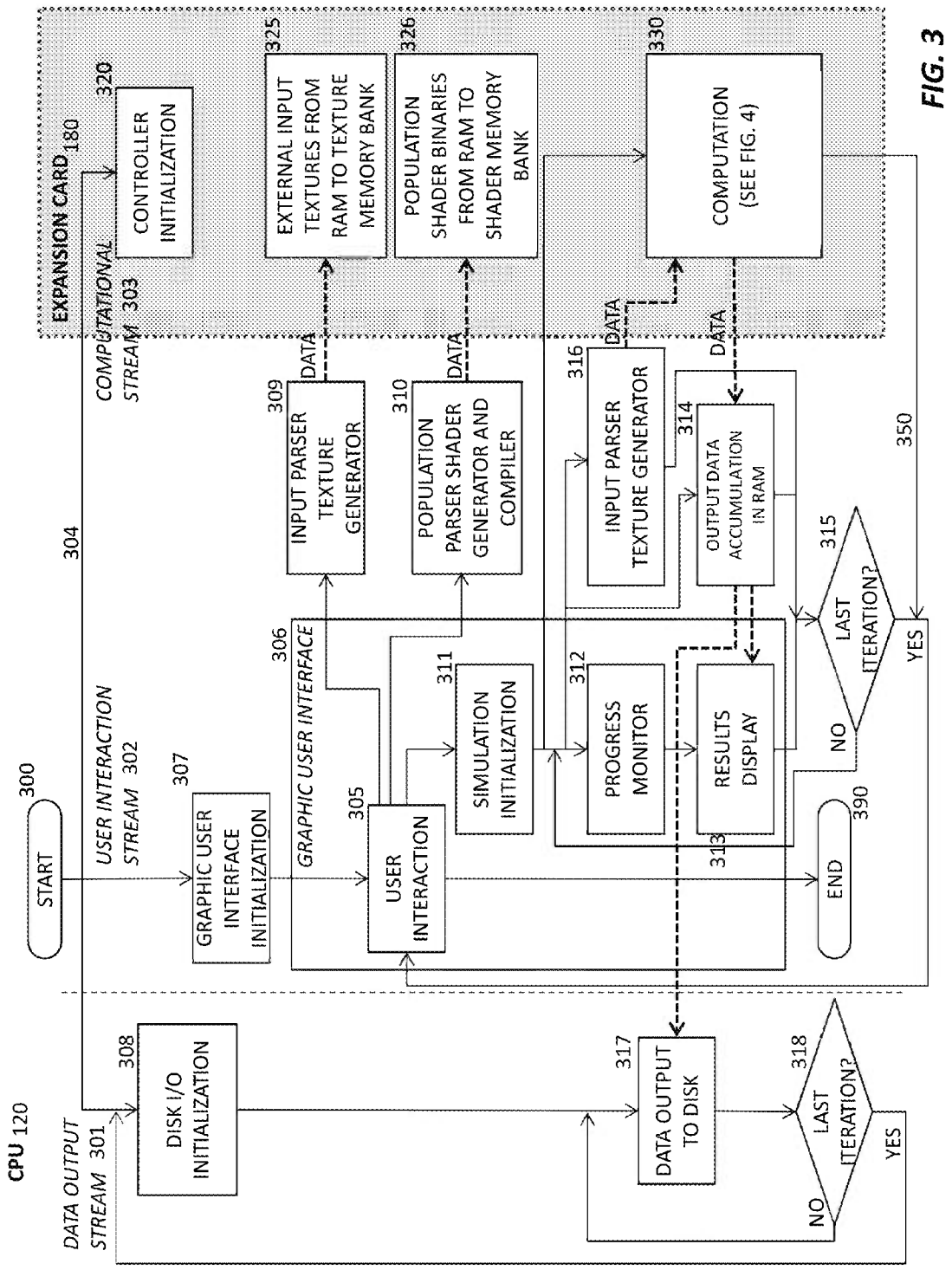

CPU 120
DATA OUTPUT STREAM 301
USER INTERACTION STREAM 302
COMPUTATIONAL STREAM 303
304
EXPANSION CARD 180
320
CONTROLLER INITIALIZATION
325
EXTERNAL INPUT TEXTURES FROM RAM TO TEXTURE MEMORY BANK
326
POPULATION SHADER BINARIES FROM RAM TO SHADER MEMORY BANK
330
COMPUTATION (SEE FIG. 4)
300
START
307
GRAPHIC USER INTERFACE INITIALIZATION
306
GRAPHIC USER INTERFACE
305
USER INTERACTION
311
SIMULATION INITIALIZATION
309
INPUT PARSER TEXTURE GENERATOR
310
POPULATION PARSER SHADER GENERATOR AND COMPILER
316
INPUT PARSER TEXTURE GENERATOR
314
OUTPUT DATA ACCUMULATION IN RAM
312
PROGRESS MONITOR
313
RESULTS DISPLAY
315
LAST ITERATION?
NO
YES
350
390
END
308
DISK I/O INITIALIZATION
317
DATA OUTPUT TO DISK
318
LAST ITERATION?
NO
YES
DATA
DATA
DATA
DATA

FIG. 3

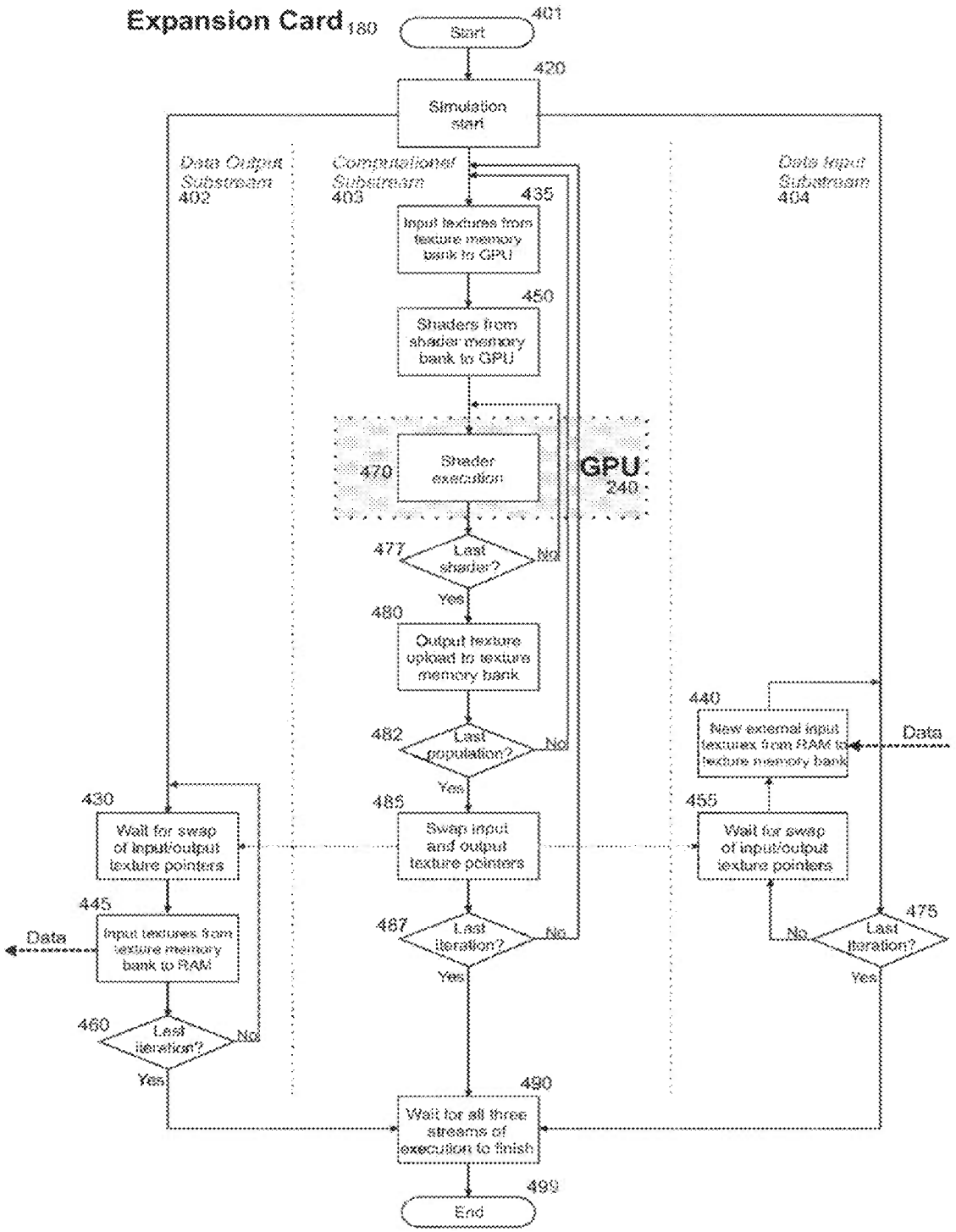

Expansion Card 180
401
Start
420
Simulation start
Data Output Substream 402
Computational Substream 403
Data Input Substream 404
435
Input textures from texture memory bank to GPU
450
Shaders from shader memory bank to GPU
470
Shader execution
GPU 240
477
Last shader?
No
Yes
480
Output texture upload to texture memory bank
482
Last population?
No
Yes
440
New external input textures from RAM to texture memory bank
Data
430
Wait for swap of input/output texture pointers
485
Swap input and output texture pointers
455
Wait for swap of input/output texture pointers
445
Input textures from texture memory bank to RAM
Data
487
Last iteration?
No
Yes
475
Last iteration?
No
Yes
460
Last iteration?
No
Yes
490
Wait for all three streams of execution to finish
499
End

FIG 4

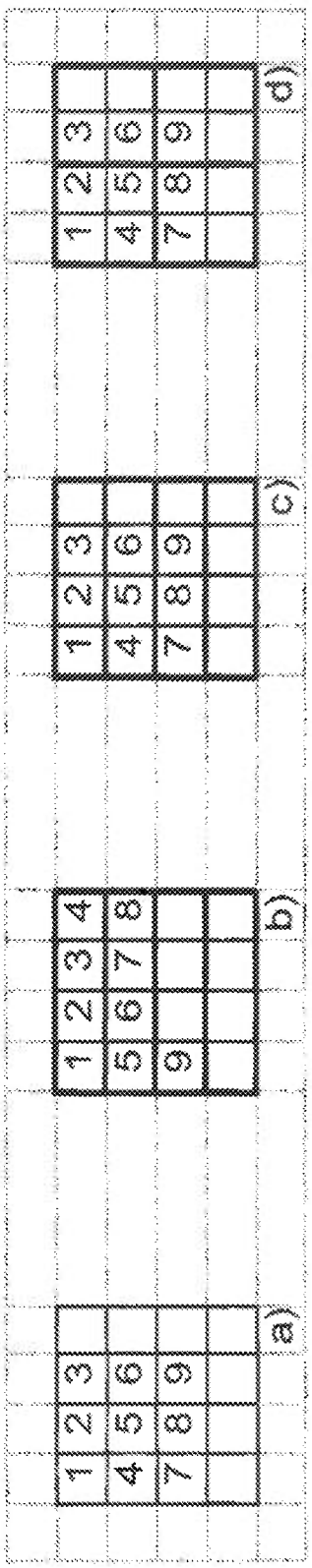

a)
b)
c)
d)
Bold lines show the packing of data into pixels with four color components.

FIG. 5

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 58249

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

	Firm or Individual Name				
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box).

Neurala, Inc.

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature		Date (Optional)	13 JULY 2016
Name	Massimiliano Versace		
Title	CEO, Neurala, Inc.		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PTO/AIA/26 (04-14)
Approved for use through 07/31/2016. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT	Docket Number (Optional) NRLA-001/02US 320730-2002

In re Application of: GORCHETCHNIKOV, Anatoli

Application No.: 14/147,015

Filed: 01/03/2014

For: GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD

The applicant, Neurala, Inc., owner of 100% percent interest in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of **prior patent** No. 8,648,867 as the term of said **prior patent** is presently shortened by any terminal disclaimer. The applicant hereby agrees that any patent so granted on the instant application shall be enforceable only for any during such period that it and the **prior patent** are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.

In making the above disclaimer, the applicant does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the **prior patent**, "as the term of said **prior patent** is presently shortened by any terminal disclaimer," in the event that said **prior patent** later:

expires for failure to pay a maintenance fee;
is held unenforceable;
is found invalid by a court of competent jurisdiction;
is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
has all claims canceled by a reexamination certificate;
is reissued; or
is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

Check either box 1 or 2 below, if appropriate.

1. ☐ The undersigned is the applicant. If the applicant is an assignee, the undersigned is authorized to act on behalf of the assignee.

I hereby acknowledge that any willful false statements made are punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

2. ☒ The undersigned is an attorney or agent of record. Reg. No. 65634

/Christopher Max Colice/ — June 30, 2015
Signature — Date

Christopher Max Colice
Typed or printed name

Title — 617 937-2362
Telephone Number

☒ Terminal disclaimer fee under 37 CFR 1.20(d) included.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

This collection of information is required by 37 CFR 1.321. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

American LegalNet, Inc.
www.FormsWorkFlow.com

PTO/AIA/96 (08-12)
Approved for use through 01/31/2013. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

Applicant/Patent Owner: Neurala, Inc.

Application No./Patent No.: 9,189,828 Filed/Issue Date: 11-17-2015

Titled: GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD

Neurala, Inc. (Name of Assignee) , a corporation (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that, for the patent application/patent identified above, it is (choose **one** of options 1, 2, 3 or 4 below):

1. [x] The assignee of the entire right, title, and interest.

2. [ ] An assignee of less than the entire right, title, and interest (check applicable box):

   [ ] The extent (by percentage) of its ownership interest is _____ %). Additional Statement(s) by the owners holding the balance of the interest must be submitted to account for 100% of the ownership interest.

   [ ] There are unspecified percentages of ownership. The other parties, including inventors, who together own the entire right, title and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

3. [ ] The assignee of an undivided interest in the entirety (a complete assignment from one of the joint inventors was made). The other parties, including inventors, who together own the entire right, title, and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

4. [ ] The recipient, via a court proceeding or the like (*e.g.*, bankruptcy, probate), of an undivided interest in the entirety (a complete transfer of ownership interest was made). The certified document(s) showing the transfer is attached.

The interest identified in option 1, 2 or 3 above (not option 4) is evidenced by either (choose **one** of options A or B below):

A. [ ] An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____ , Frame _____ , or for which a copy thereof is attached.

B. [x] A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Inventors To: Neurala LLC
   The document was recorded in the United States Patent and Trademark Office at
   Reel 032138 , Frame 0737 , or for which a copy thereof is attached.

2. From: Neurala LLC To: Neurala, Inc.
   The document was recorded in the United States Patent and Trademark Office at
   Reel 032166 , Frame 0594 , or for which a copy thereof is attached.

153829963 v1

PTO/AIA/96 (08-12)
Approved for use through 01/31/2013. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

3. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____ , Frame _____ , or for which a copy thereof is attached.

4. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____ , Frame _____ , or for which a copy thereof is attached.

5. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____ , Frame _____ , or for which a copy thereof is attached.

6. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____ , Frame _____ , or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).

☒ As required by 37 CFR 3.73(c)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

/Christopher Max Colice/	November 9, 2017
Signature	Date
Christopher Max Colice	65,634
Printed or Typed Name	Title or Registration Number

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NRLA-001/03US 320730-2040
	Application Number	
Title of Invention	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2:

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Anatoli		Gorchetchnikov	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Belmont	State/Province	MA	Country of Residence [i]	US

Mailing Address of Inventor:

Address 1	15 Jefferson Street		
Address 2			
City	Newton	State/Province	MA
Postal Code	02458	Country [i]	US

Inventor 2

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Heather	Marie	Ames	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Milton	State/Province	MA	Country of Residence [i]	US

Mailing Address of Inventor:

Address 1	13 Myers Lane		
Address 2			
City	Milton	State/Province	MA
Postal Code	02186	Country [i]	US

Inventor 3

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Massimiliano		Versace	

PTO/AIA/14 (08-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NRLA-001/03US 320730-2040
	Application Number	
Title of Invention	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD	

Residence Information (Select One)	◉ US Residency	○ Non US Residency	○ Active US Military Service		
City	Milton	**State/Province**	MA	**Country of Residence** i	US

Mailing Address of Inventor:

Address 1	13 Myers Lane		
Address 2			
City	Milton	**State/Province**	MA
Postal Code	02186	**Country** i	US

Inventor 4

Legal Name

Prefix	**Given Name**	**Middle Name**	**Family Name**	**Suffix**
	Fabrizio		Santini	

Residence Information (Select One)	◉ US Residency	○ Non US Residency	○ Active US Military Service		
City	Jamaica Plain	**State/Province**	MA	**Country of Residence** i	US

Mailing Address of Inventor:

Address 1	22 Sigourney St. Unit E		
Address 2			
City	Jamaica Plain	**State/Province**	MA
Postal Code	02130	**Country** i	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

☐ **An Address is being provided for the correspondence Information of this application.**

Customer Number	58249		
Email Address	zPATDCDocketing@cooley.com	Add Email	Remove Email

Application Information:

Title of the Invention	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD		
Attorney Docket Number	NRLA-001/03US 320730-2040	**Small Entity Status Claimed**	☒
Application Type			
Subject Matter			
Total Number of Drawing Sheets (if any)	5	**Suggested Figure for Publication (if any)**	

PTO/AIA/14 (08-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NRLA-001/03US 320730-2040
	Application Number	
Title of Invention	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD	

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	◉ Customer Number	○ US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	58249		

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix	Remove
Registration Number					
Prefix	**Given Name**	Middle Name	**Family Name**	Suffix	Remove
Registration Number					

Additional Representative Information blocks may be generated within this form by selecting the **Add** button.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NRLA-001/03US 320730-2040
	Application Number	
Title of Invention	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD	

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing this information in the application data sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the application number blank.

Prior Application Status	Patented	Remove

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
	reissued of	14147015	2014-01-03	9189828	2015-11-17

Prior Application Status	Patented	Remove

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
14147015	Continuation of	11860254	2007-09-24	8648867	2014-02-11

Prior Application Status	Expired	Remove

Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
11860254	Claims benefit of provisional	60826892	2006-09-25

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button.

Foreign Priority Information:

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX) the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Application Number	Country	Filing Date (YYYY-MM-DD)	Access Code (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button.

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

PTO/AIA/14 (08-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NRLA-001/03US 320730-2040
	Application Number	
Title of Invention	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD	

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

PTO/AIA/14 (08-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NRLA-001/03US 320730-2040
	Application Number	
Title of Invention	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD	

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

PTO/AIA/14 (08-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NRLA-001/03US 320730-2040
	Application Number	
Title of Invention	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD	

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

◉ Assignee	○ Legal Representative under 35 U.S.C. 117	○ Joint Inventor
○ Person to whom the inventor is obligated to assign.	○ Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:	

If the Applicant is an Organization check here. ☒

Organization Name	Neurala, Inc.

Mailing Address Information For Applicant:

Address 1	51 Sleeper St, 3rd Floor		
Address 2			
City	Boston	**State/Province**	MA
Country[i]	US	Postal Code	02210
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button.

PTO/AIA/14 (08-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NRLA-001/03US 320730-2040
	Application Number	
Title of Invention	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD	

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country i		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button.

PTO/AIA/14 (08-15)
Approved for use through 04/30/2017. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	NRLA-001/03US 320730-2040
	Application Number	
Title of Invention	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD	

Signature:

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e. g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/Christopher Max Colice/			Date (YYYY-MM-DD)	
First Name	Christopher	Last Name	Colice	Registration Number	65634

Additional Signature may be generated within this form by selecting the Add button.

PTO/AIA/05 (06-12)
Approved for use through 08/31/2013. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REISSUE APPLICATION DECLARATION BY THE INVENTOR	Docket Number (Optional) NRLA-001/03US 320730-2040

I hereby declare that:
Each inventor's residence and mailing address are stated below next to their name.
I believe I am the original inventor or an original joint inventor of the subject matter which is described and claimed in patent number 9,189,828, granted November 17, 2015 and for which a reissue patent is sought on the invention titled Graphic Processor Based Accelerator System and Method,

_____;

the specification of which

☑ is attached hereto.

☐ was filed on _____ as reissue application number _____ .

The above-identified application was made or authorized to be made by me.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

I believe the original patent to be wholly or partly inoperative or invalid, for the reasons described below. (Check all boxes that apply.)

☐ by reason of a defective specification or drawing.

☑ by reason of the patentee claiming more or less than he had the right to claim in the patent.

☑ by reason of other errors.

At least one error upon which reissue is based is described below. If the reissue is a broadening reissue, a claim that the application seeks to broaden must be identified:

Reissue is requested because of antecedent basis errors in issued claim 1 of US 9,189,828 and because Applicant erred by claiming less than it had a right to claim in US 9,189,828. The amendments to claim 1 correct the antecedent basis errors and the new claims, including claim 21, correct Applicant's error in claiming less than it had a right to claim.

This collection of information is required by 37 CFR 1.175. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

(REISSUE APPLICATION DECLARATION BY THE INVENTOR, page 2)

Docket Number (Optional): NRLA-001/03US 320730-2040

Note: To appoint a power of attorney, use form PTO/AIA/81

Correspondence Address: Direct all communications about the application to:

☑ The address associated with Customer Number: 58249

OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

Legal name of sole or first inventor (E.g., Given Name (first and middle (if any)) and Family Name or Surname)

Massimiliano Versace

Inventor's Signature		Date (Optional)
Residence: City Milton	State MA	Country USA

Mailing Address

13 Myers Lane

City	State	Zip	Country
Milton	MA	02186	USA

Additional joint inventors are named on the _____ supplemental sheet(s) PTO/AIA/02 attached hereto

PTO/AIA/05 (06-12)
Approved for use through 08/31/2013. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REISSUE APPLICATION DECLARATION BY THE INVENTOR	Docket Number (Optional) NRLA-001/03US 320730-2040

I hereby declare that:
Each inventor's residence and mailing address are stated below next to their name.
I believe I am the original inventor or an original joint inventor of the subject matter which is described and claimed in patent number 9,189,828, granted November 17, 2015 and for which a reissue patent is sought on the invention titled Graphic Processor Based Accelerator System and Method,

_____;

the specification of which

☑ is attached hereto.

☐ was filed on _____ as reissue application number _____ .

The above-identified application was made or authorized to be made by me.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

I believe the original patent to be wholly or partly inoperative or invalid, for the reasons described below. (Check all boxes that apply.)

☐ by reason of a defective specification or drawing.

☑ by reason of the patentee claiming more or less than he had the right to claim in the patent.

☑ by reason of other errors.

At least one error upon which reissue is based is described below. If the reissue is a broadening reissue, a claim that the application seeks to broaden must be identified:

Reissue is requested because of antecedent basis errors in issued claim 1 of US 9,189,828 and because Applicant erred by claiming less than it had a right to claim in US 9,189,828. The amendments to claim 1 correct the antecedent basis errors and the new claims, including claim 21, correct Applicant's error in claiming less than it had a right to claim.

This collection of information is required by 37 CFR 1.175. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

(REISSUE APPLICATION DECLARATION BY THE INVENTOR, page 2)

Docket Number (Optional): NRLA-001/03US 320730-2040

Note: To appoint a power of attorney, use form PTO/AIA/81

Correspondence Address. Direct all communications about the application to:

☑ The address associated with Customer Number: 58249

OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

Legal name of sole or first inventor (E.g., Given Name (first and middle (if any)) and Family Name or Surname)

Anatoli Gorchetchnikov

Inventor's Signature		Date (Optional)
Residence: City: Newton	State: MA	Country: USA

Mailing Address:

15 Jefferson Street

City: Newton	State: MA	Zip: 02458	Country: USA

1 Additional joint inventors are named on the _____ supplemental sheet(s) PTO/AIA/05 attached hereto

NVIDIA Ex1002 Page 42 of 1081

PTO/AIA/05 (06-12)
Approved for use through 08/31/2013. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REISSUE APPLICATION DECLARATION BY THE INVENTOR	Docket Number (Optional) NRLA-001/03US 320730-2040

I hereby declare that:
Each inventor's residence and mailing address are stated below next to their name.
I believe I am the original inventor or an original joint inventor of the subject matter which is described and claimed in patent number 9,189,828, granted November 17, 2015 and for which a reissue patent is sought on the invention titled Graphic Processor Based Accelerator System and Method,

;

the specification of which

☑ is attached hereto.

☐ was filed on _____ as reissue application number _____ .

The above-identified application was made or authorized to be made by me.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

I believe the original patent to be wholly or partly inoperative or invalid, for the reasons described below. (Check all boxes that apply.)

☐ by reason of a defective specification or drawing.

☑ by reason of the patentee claiming more or less than he had the right to claim in the patent.

☑ by reason of other errors.

At least one error upon which reissue is based is described below. If the reissue is a broadening reissue, a claim that the application seeks to broaden must be identified:

Reissue is requested because of antecedent basis errors in issued claim 1 of US 9,189,828 and because Applicant erred by claiming less than it had a right to claim in US 9,189,828. The amendments to claim 1 correct the antecedent basis errors and the new claims, including claim 21, correct Applicant's error in claiming less than it had a right to claim.

This collection of information is required by 37 CFR 1.175. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/AIA/05 (08-12)
Approved for use through 08/31/2013. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

(REISSUE APPLICATION DECLARATION BY THE INVENTOR, page 2)

Docket Number (Optional): NRLA-001A03US 320730-2040

Note: To appoint a power of attorney, use form PTO/AIA/81.

Correspondence Address: Direct all communications about the application to:

☑ The address associated with Customer Number: 58249

OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

Legal name of sole or first inventor (E.g., Given Name (first and middle (if any)) and Family Name or Surname)

Heather Marie Ames

Inventor's Signature	Date (Optional)
	

Residence: City	State	Country
Milton	MA	USA

Mailing Address

13 Myers Lane

City	State	Zip	Country
Milton	MA	02186	USA

Additional joint inventors are named on this _____ supplemental sheet(s) PTO/AIA/18 attached hereto

NVIDIA Ex1002 Page 44 of 1081

PTO/AIA/05 (06-12)
Approved for use through 08/31/2013. OMB 0651-0033
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

REISSUE APPLICATION DECLARATION BY THE INVENTOR	Docket Number (Optional) NRLA-001/03US 320730-2040

I hereby declare that:
Each inventor's residence and mailing address are stated below next to their name.
I believe I am the original inventor or an original joint inventor of the subject matter which is described and claimed in patent number 9,189,828, granted November 17, 2015 and for which a reissue patent is sought on the invention titled Graphic Processor Based Accelerator System and Method,

_____;

the specification of which

☑ is attached hereto.

☐ was filed on _____ as reissue application number _____ .

The above-identified application was made or authorized to be made by me.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

I believe the original patent to be wholly or partly inoperative or invalid, for the reasons described below. (Check all boxes that apply.)

☐ by reason of a defective specification or drawing.

☑ by reason of the patentee claiming more or less than he had the right to claim in the patent.

☑ by reason of other errors.

At least one error upon which reissue is based is described below. If the reissue is a broadening reissue, a claim that the application seeks to broaden must be identified:

Reissue is requested because of antecedent basis errors in issued claim 1 of US 9,189,828 and because Applicant erred by claiming less than it had a right to claim in US 9,189,828. The amendments to claim 1 correct the antecedent basis errors and the new claims, including claim 21, correct Applicant's error in claiming less than it had a right to claim.

This collection of information is required by 37 CFR 1.175. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

(REISSUE APPLICATION DECLARATION BY THE INVENTOR, page 2)

Docket Number (Optional)
NRLA-001/03US 320730-2040

Note: To appoint a power of attorney, use form PTO/AIA/81

Correspondence Address: Direct all communications about the application to:

☑ The address associated with Customer Number: 58249

OR

☐ Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

Legal name of sole or first inventor (*E.g.*, Given Name (first and middle (if any) and Family Name or Surname)

Fabrizio Santini

Inventor's Signature		Date (Optional)	
Residence: City	State	Country	
Mailing Address			
City	State	Zip	Country

☐ Additional joint inventors are named on the _____ supplemental sheet(s) PTO/AIA/10 attached herein

REMARKS

Patentee is filing this broadening reissue application to correct antecedent basis errors in claim 1 and to correct Applicant's error in claiming less than it had a right to claim in the parent case, U.S. Patent No. 9,189,828 ("the '828 patent"). Two examples of Applicant claiming less than it had a right to claim are that no independent claim is directed to "[a]n accelerator system" as in new claims 21 or "[a] method of performing a sequence of computations representing an artificial neural network" as in new claim 42. The new claims correct Applicant's error in claiming less than it had a right to claim. And the amendments to claim 1 correct the antecedent basis errors.

Statement of Status of Claims and Support for Claim Changes per 37 C.F.R. 1.173(c)

Claim 1 is being amended to correct antecedent basis errors. Support for the amendments to claim 1 appear in claim itself.

Claims 2–20 are not being amended (i.e., they remain as issued).

Claims 21–136 are being added. Support for new claims 21–136 appears at least in the following sections of the '828 patent: column 3, lines 59–65; column 4, lines 8–24; column 4, lines 46–50; column 5, lines 1–12; column 5, lines 17–52; column 6, lines 7–12; column 7, lines 1–53; column 7, line 66, through column 8, line 2; column 8, lines 12–17; column 8, lines 25–28; claims 1–20; and FIGS. 1–3.

The Director is hereby authorized to charge any appropriate fees under 37 C.F.R. §§ 1.16, 1.17, and 1.21 that may be required by this paper, and to credit any overpayment, to Deposit Account No. 50-1283.

Dated: November 9, 2017

Respectfully submitted,
COOLEY LLP

COOLEY LLP
ATTN: IP Docketing Department
1299 Pennsylvania Avenue NW, Suite 700
Washington, DC 20004

Tel: (617) 937-2362
Fax: (202) 842-7899

By: /Christopher Max Colice/
Christopher Max Colice
Reg. No. 65634

IN THE CLAIMS:

Set forth below in ascending order, with status identifiers, is a complete listing of all claims currently under examination. Changes to any amended claims are indicated by strikethrough and underlining. This listing also reflects any cancellation and/or addition of claims.

1. (Amended) A computer system, comprising:

a central processing unit to receive input data;

main memory, operably coupled to the central processing unit via a bus, to store the input data received by the central processing unit;

an accelerator, operably coupled to the central processing unit and the ~~first~~ main memory via the bus, to receive at least a portion of the input data from the main memory, the accelerator comprising:

at least one graphics processing unit to perform a sequence of computations on the at least a portion of the input data so as to generate output data, intermediate computations in the sequence of computations yielding intermediate results; and

accelerator memory, operably coupled to the at least one graphic processing unit, to store the results of the ~~plurality of sequential~~ sequence of computations; and

a controller, operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory and to transfer at least a portion of the output data from the accelerator memory to the main memory during performance of the sequence of computations by the at least one graphic processing unit.

2. The computer system of claim 1, wherein the central processing unit is configured to receive the input data in response to a user interaction.

3. The computer system of claim 1, wherein:

the central processing unit is configured to receive the input data at a first rate; and

the at least one graphics processing unit is configured to perform the sequence of computations at a second rate different than the first rate.

4. The computer system of claim 1, wherein the main memory is configured to store a copy of the output data stored in the accelerator memory.

5. The computer system of claim 1, wherein an output of at least one computation in the sequence of computations represents an output of at least one neuron in an artificial neural network.

6. The computer system of claim 1, wherein accelerator memory comprises:

a first memory bank to store parameters common to all of the computations in the sequence of computations; and

a second memory bank to store data specific to at least one computation in the sequence of computations.

7. The computer system of claim 1, wherein the controller is configured to transfer the output data from the accelerator memory to the main memory without transferring any of the intermediate results from the accelerator memory to the main memory so as to reduce data transfer via the bus.

8. The computer system of claim 1, wherein the controller is configured to transfer at least a portion of the output data from the accelerator memory to the main memory after the at least one graphics processing unit has begun to perform another sequence of computations.

9. The computer system of claim 8, wherein the controller is configured to initiate transfer of the at least a portion of the input data and to transfer the at least a portion of the output data in parallel with performance of at least one computation in the other sequence of computations by the at least one graphics processing unit.

10. The computer system of claim 1, wherein the controller is configured to control execution of the sequence of computations by the at least one graphics processing unit.

11. The computer system of claim 1, further comprising:

at least one of a video camera, a microphone, or a cell recording electrode, operably coupled to the central processor unit, to acquire the input data in real time.

12. A method of performing a sequence of computations on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory, the method comprising:

(A) performing, by the GPU, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data, intermediate computations in the sequence of computations yielding intermediate results;

(B) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the input data from the main memory to the accelerator via the bus; and

(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus.

13. The method of claim 12, further comprising:

storing the input data in the main memory in response to a user interaction.

14. The method of claim 12, further comprising:

receiving the input data at a first rate; and

wherein (A) comprises performing the sequence of computations at a second rate different than the first rate.

15. The method of claim 12, wherein (A) comprises:

generating an output representative of an output of at least one neuron in an artificial neural network.

16. The method of claim 12, wherein (C) comprises:

transferring the second portion of the output data from the accelerator memory to the main memory without transferring any of the intermediate results of the plurality of sequential computations from the accelerator memory to the main memory so as to reduce data transfer via the bus.

17. The method of claim 12, wherein (C) comprises:

transferring the second portion of the output data from the accelerator memory to the main memory after the GPU has begun to perform another sequence of computations.

18. The method of claim 17, wherein (C) further comprises:

initiating transfer of the second portion of the output data in parallel with performance of at least one computation in the other sequence of computations.

19. The method of claim 12, further comprising:

acquiring the input data in real time with at least one of a video camera, a microphone, or a cell recording electrode operably coupled to the CPU.

20. The method of claim 12, further comprising:

storing parameters common to all of the computations in the sequence of computations in a first memory bank in the accelerator memory; and

storing data specific to at least one computation in the sequence of computations in a second memory bank in the accelerator memory.

21. (New) An accelerator system comprising:

a graphics processing unit (GPU) to perform a sequence of computations on input data so as to generate output data, an output of a computation in the sequence of computations representing an output of a neuron in an artificial neural network;

a memory, operably coupled to the GPU, to store the input data and the output data; and

a controller, operably coupled to the GPU and the memory, to transfer the input data into the memory and to transfer at least a portion of the output data from the memory during performance of the sequence of computations by the GPU.

22. (New) The accelerator system of claim 21, wherein the GPU is operably coupled to a central processing unit (CPU) of a computer system.

23. (New) The accelerator system of claim 21, wherein the GPU is mounted on an expansion card operably coupled to a motherboard of a computer system.

24. (New) The accelerator system of claim 21, further comprising:

a central processing unit (CPU), a system memory, and a non-volatile memory installed on a motherboard.

25. (New) The accelerator system of claim 24, wherein the GPU is operably coupled to the CPU via a system bus on the motherboard.

26. (New) The accelerator system of claim 25, wherein the system bus has a bandwidth of 3.2 GB/s to 15 GB/s.

27. (New) The accelerator system of claim 25, wherein the controller is configured to transfer the input data and the at least a portion of the output data via the system bus.

28. (New) The accelerator system of claim 25, further comprising:

a video card, operably coupled to the motherboard via a local bus, to transfer graphical information to a display.

29. (New) The accelerator system of claim 28, wherein the local bus has a lower bandwidth than the system bus.

30. (New) The accelerator system of claim 21, wherein the memory comprises:

a first memory partition to store parameters common to every computation in the sequence of computations.

31. (New) The accelerator system of claim 30, wherein the memory further comprises:

a second memory partition to store data specific to the computation in the sequence of computations.

32. (New) The accelerator system of claim 31, wherein the second memory partition comprises:

a first partition to hold external input data patterns;

a second partition to hold representations of internal variables;

a third partition to hold an input of the computation in the sequence of computations; and

a fourth partition to hold the output of the computation in the sequence of computations.

33. (New) The accelerator system of claim 31, wherein the second memory partition is configured to accumulate the output of the computation in the sequence of computations during execution, by the GPU, of computational elements of the computation in the sequence of computations.

34. (New) The accelerator system of claim 31, wherein the second memory partition is configured to store an output of a previous computation in the sequence of computations, the output of the previous computation being accessible to computational elements executed by the GPU during the computation in the sequence of computations.

35. (New) The accelerator system of claim 21, wherein the computation in the sequence of computations performed by the GPU comprises a plurality of computational elements.

36. (New) The accelerator system of claim 35, wherein the plurality of computational elements represents a layer of neurons in an artificial neural network.

37. (New) The accelerator system of claim 36, wherein all neurons in the layer of neurons are described by the same equation.

38. (New) The accelerator system of claim 21, wherein the memory is configured to store intermediate results of the sequence of computations and the controller is configured to transfer the output data from the memory to a main memory without transferring the intermediate results.

39. (New) The accelerator system of claim 21, wherein the controller is configured to initiate transfer of at least a portion of the input data and to transfer the at least a portion of the output data in parallel.

40. (New) The accelerator system of claim 21, wherein the input data is acquired by at least one of a video camera, a microphone, or a cell recording electrode.

41. (New) The accelerator system of claim 21, wherein the input data is loaded from disk.

42. (New) A method of performing a sequence of computations representing an artificial neural network, the method comprising:

performing, by a graphics processing unit (GPU), a first computation in the sequence of computations on first input data to generate first output data, an output of a computation in the sequence of computations representing an output of a neuron in the artificial neural network;

storing, in a memory operably coupled to the GPU, the first input data and the first output data; and

transferring second input data into the memory during performance of the sequence of computations by the GPU.

43. (New) The method of claim 42, wherein transferring the second input data comprises transferring the second input data via a bus operably coupled to a central processing unit (CPU).

44. (New) The method of claim 42, further comprising:

transferring the first output data from the memory to another memory during a second computation in the sequence of computations.

45. (New) The method of claim 44, further comprising:

storing intermediate results of the sequence of computations in the memory, and

wherein transferring the first output data from the memory to the other memory occurs without transferring the intermediate results of the sequence of computations.

46. (New) The method of claim 44, wherein transferring the second input data and transferring the first output data occurs in parallel.

47. (New) The method of claim 42, further comprising:

storing, in a first memory partition of the memory, parameters common to all of the computations in the sequence of computations.

48. (New) The method of claim 47, further comprising:

storing, in a second memory partition of the memory, data specific to the first computation in the sequence of computations.

49. (New) The method of claim 48, further comprising:

storing, in the second memory partition, external input data patterns, representations of internal variables, an input of the computation in the sequence of computations, and the output of the computation in the sequence of computations.

50. (New) The method of claim 42, wherein storing the first output data comprises:

accumulating, in the memory, outputs of computational elements executed by the GPU in performing the first computation in the sequence of computations.

51. (New) The method of claim 42, further comprising:

storing, in the memory, an output of a previous computation in the sequence of computations; and

accessing, by the GPU, the output of the previous computation during performance of the computation in the sequence of computations.

52. (New) The method of claim 42, wherein performing the first computation comprises executing a plurality of computational elements representing a layer of neurons in an artificial neural network.

53. (New) The method of claim 52, wherein all neurons in the layer of neurons are described by the same equation.

54. (New) The method of claim 42, further comprising:

acquiring the input data with at least one of a video camera, a microphone, or a cell recording electrode.

55. (New) The method of claim 42, further comprising:

loading the input data from from disk.

56. (New) An accelerator system comprising:

a graphics processing unit (GPU) to perform a sequence of computations on input data so as to generate output data, a computation in the sequence of computations comprising a plurality of computational elements, the plurality of computational elements representing a layer of neurons in an artificial neural network, an output of the computation in the sequence of computations representing an output of a neuron in the artificial neural network; and

a memory, operably coupled to the GPU, to store parameters common to all of the computations in the sequence of computations and data specific to the computation in the sequence of computations.

57. (New) The accelerator system of claim 56, wherein all neurons in the layer of neurons are described by the same equation.

58. (New) The accelerator system of claim 56, wherein the memory is configured to hold:
external input data patterns;
representations of internal variables;
an input to the computation in the sequence of computations; and
the output of the computation in the sequence of computations.

59. (New) The accelerator system of claim 56, wherein the memory is configured to accumulate the output of the computation in the sequence of computations during execution of the plurality of computational elements by the GPU.

60. (New) The accelerator system of claim 56, wherein the memory is configured to store an output of a previous computation in the sequence of computations, the output of the previous computation being accessible to the plurality of computational elements during the computation in the sequence of computations.

61. (New) The accelerator system of claim 56, further comprising:
a controller, operably coupled to the GPU and the memory, to transfer the input data into the memory and to transfer at least a portion of the output data from the memory during performance of the sequence of computations by the GPU.

62. (New) A method comprising:
performing, with a graphics processing unit (GPU), a sequence of computations on input data to generate output data, a computation in the sequence of computations comprising a plurality of computational elements, the plurality of computational elements representing a layer of neurons in an artificial neural network, an output of the computation in the sequence of computations representing an output of a neuron in the artificial neural network;

storing, in a memory operably coupled to the GPU, parameters common to all of the computations in the sequence of computations; and

storing, in the memory, data specific to the computation in the sequence of computations.

63. (New) The method of claim 62, wherein all neurons in the layer of neurons are described by the same equation.

64. (New) The method of claim 62, further comprising:

storing, in the memory, external input data patterns;

storing, in the memory, representations of internal variables; and

storing, in the memory, an input of the computation in the sequence of computations.

65. (New) The method of claim 62, further comprising:

accumulating, in the memory, the output of the computation in the sequence of computations during execution of the plurality of computational elements by the GPU.

66. (New) The method of claim 62, further comprising:

storing, in the memory, an output of a previous computation in the sequence of computations; and

accessing, by the GPU, the output of the previous computation during the computation in the sequence of computations.

67. (New) The method of claim 62, further comprising:

transferring the input data into the memory and at least a portion of the output data from the memory during performance of the sequence of computations by the GPU.

68. (New) A method of executing computations representing an artificial neural network on a computer system comprising at least one central processing unit (CPU), a processing unit, a first memory partition, and a second memory partition, the method comprising:

executing, by the at least one CPU, a user interaction stream, the user interaction stream controlling transfer of inputs to the first memory partition and the second memory partition;

executing, by the processing unit, a computational stream, the computational stream controlling data exchange between the user interaction stream and the computational stream during execution of the computations representing the artificial neural network.

69. (New) The method of claim 68, wherein the user interaction stream controls data exchange between the user interaction stream and the computational stream outside of execution of the computations representing the artificial neural network.

70. (New) The method of claim 68, wherein executing the user interaction stream comprises:

controlling setting and editing of computational elements of the computations representing the artificial neural network.

71. (New) The method of claim 68, wherein executing the user interaction stream comprises:

controlling setting and editing of parameters of the computations representing the artificial neural network.

72. (New) The method of claim 68, wherein executing the user interaction stream comprises:

controlling setting and editing of parameters of the inputs to the artificial neural network.

73. (New) The method of claim 68, wherein executing the user interaction stream comprises:

specifying an output to be saved to disk and/or displayed on a screen.

74. (New) The method of claim 68, wherein executing the user interaction stream comprises:

parsing elements to be used in the computations representing the artificial neural network.

75. (New) The method of claim 74, wherein the processing unit comprises a graphics processing unit (CPU) and executing the user interaction stream further comprises:

converting the elements into GPU programs.

76. (New) The method of claim 75, wherein executing the user interaction stream comprises:

compiling the GPU programs.

77. (New) The method of claim 76, wherein executing the user interaction stream comprises:

transferring the GPU programs to the second memory partition.

78. (New) The method of claim 68, further comprising:

executing, by the at least one CPU, a data output stream, the data output stream controlling transfer of outputs of the computations representing the artificial neural network to disk.

79. (New) A system for executing computations representing an artificial neural network, the system comprising:

at least one memory;

at least one central processing unit (CPU), operably coupled to the at least one memory, to execute a user interaction stream, the user interaction stream controlling transfer of inputs to the at least one memory; and

a processing unit, operably coupled to the at least one memory and the at least one CPU, to execute a computational stream, the computational stream controlling data exchange between the user interaction stream and the computational stream during execution of the computations representing the artificial neural network.

80. (New) The system of claim 79, wherein the user interaction stream controls data exchange between the user interaction stream and the computational stream outside of execution of the computations representing the artificial neural network.

81. (New) The system of claim 79, wherein the user interaction stream controls setting and editing of computational elements of the computations representing the artificial neural network.

82. (New) The system of claim 79, wherein the user interaction stream controls setting and editing of parameters of the computations representing the artificial neural network.

83. (New) The system of claim 79, wherein the user interaction stream controls setting and editing of parameters of the inputs to the artificial neural network.

84. (New) The system of claim 79, wherein the user interaction stream specifies an output to be saved to disk and/or displayed on a screen.

85. (New) The system of claim 79, wherein the user interaction stream parses elements to be used in the computations representing the artificial neural network.

86. (New) The system of claim 85, wherein the processing unit comprises a graphics processing unit (GPU) and the user interaction stream converts the elements into GPU programs.

87. (New) The system of claim 86, wherein the user interaction stream compiles the GPU programs.

88. (New) The system of claim 87, wherein the user interaction stream transfers the GPU programs to a shader memory partition of the at least one memory.

89. (New) The system of claim 68, wherein the at least one CPU is further configured to execute a data output stream, the data output stream controlling transfer of outputs of the computations representing the artificial neural network to disk.

90. (New) A method of performing a sequence of computations with a processing unit, the method comprising:

receiving input data;

performing, by the processing unit, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data, intermediate computations in the sequence of computations yielding intermediate results, the first portion of the output data representing an output of an artificial neural network;

in parallel with performing the sequence of computations by the processing unit, transferring a second portion of the input data from a first memory partition to a second memory partition; and

in parallel with performing the sequence of computations by the processing unit, transferring a second portion of the output data from the second memory partition to the first memory partition.

91. (New) The method of claim 90, wherein receiving the input data comprises loading the input data from disk.

92. (New) The method of claim 90, wherein receiving the input data comprises receiving the input data from at least one of a video camera, a microphone, or a cell recording electrode.

93. (New) The method of claim 90, wherein receiving the input data occurs at a first rate and performing the sequence of computations occurs at a second rate different than the first rate.

94. (New) The method of claim 90, wherein transferring the second portion of the output data comprises transferring the second portion of the output data from the second memory partition to the first memory partition without transferring any of the intermediate results from the second memory partition to the first memory partition.

95. (New) A system for performing a sequence of computations, the system comprising:

a first memory partition;

a second memory partition operably coupled to the first memory partition; and

a processing unit, operably coupled to the first memory partition and the second memory partition, to perform the sequence of computations on a first portion of the input data so as to generate a first portion of the output data, intermediate computations in the sequence of computations yielding intermediate results, the first portion of the output data representing an output of an artificial neural network,

wherein the first memory partition is configured to transfer a second portion of the input data to the second memory partition in parallel with performance the sequence of computations by the processing unit, and

wherein the second memory partition is configured to transfer a second portion of the output data to the first memory partition in parallel with performance the sequence of computations by the processing unit.

96. (New) The system of claim 95, wherein the first memory partition and the second memory partition are logical partitions.

97. (New) The system of claim 95, wherein the processing unit is comprises a graphics processing unit (GPU).

98. (New) The system of claim 95, wherein the processing unit is configured to receive the input data at a first rate and to perform the sequence of computations at a second rate is different than the first rate.

99. (New) The system of claim 95, wherein the second memory partition is configured to transfer the second portion of the output data to the first memory partition without transferring any of the intermediate results to the first memory partition.

100. (New) The system of claim 95, further comprising

at least one of a video camera, a microphone, or a cell recording electrode to generate the input data.

101. (New) A method for performing a numerical simulation over a plurality of computational cycles including at least a first computational cycle and a second computational cycle, the method comprising:

storing, in a first partition of a memory, first output data during the second computational cycle, the first output data generated during the first computational cycle and being referenced by a first pointer;

performing, by a processing unit operably coupled to the memory, at least one calculation on the first output data during the second computational cycle, the at least one calculation yielding second output data;

storing, in a second partition of the memory, the second output data, the second output data being referenced by a second pointer; and

swapping the first pointer and the second pointer at the conclusion of the second computational cycle such that the second output data becomes an input for a third computational cycle of the plurality of computational cycles.

102. (New) The method of claim 101, further comprising:

transferring input data into the memory from another memory coupled to a central processing unit (CPU) before the first computational cycle.

103. (New) The method of claim 101, further comprising:

providing, to a central processing unit (CPU), first output data generated during the first computational cycle after a conclusion of the first computational cycle.

104. (New) The method of claim 101, further comprising:

transferring the first output data from the memory to another memory during the second computational cycle.

105. (New) The method of claim 101, further comprising:

displaying, with a video system, an indication of the numerical simulation.

106. (New) The method of claim 101, further comprising:

dictating, to the processing unit, an order of execution of instructions.

107. (New) The method of claim 101, wherein performing the at least one calculation comprises receiving, by the processing unit, at least the portion of the input data and a series of instructions and executing, by the processing unit, the instructions on the at least the portion of the input data.

108. (New) A method of performing a numerical simulation, the method comprising:

performing, by a processing unit, at least one computation during a first computational cycle of the numerical simulation;

storing, by a memory operably coupled to the processing unit, data used to perform the at least one computation;

controlling performance of the at least one computation by the processing unit;

controlling transfer of the output data from the processing unit to the memory during the first computational cycle; and

controlling transfer of input data to the memory for use by the processing unit during a second computational cycle of the numerical simulation.

109. (New) The method of claim 108, further comprising:

dictating an order of execution of instructions to the processing unit.

110. (New) The method of claim 108, further comprising:

transferring output data from the memory to a main memory while the processing unit performs the at least one computation.

111. (New) The method of claim 108, further comprising:

perform successive computational cycles of the numerical simulation by feeding output data generated by the processing unit from a previous computational cycle and input data for a next computational cycle into the processing unit.

112. (New) A system for performing a numerical simulation on input data, the system comprising:

a central processing unit (CPU) to provide first input data;

a memory, operably coupled to the CPU, to store the first input data in a first partition, referenced by first pointer, before a first computational cycle of the numerical simulation;

a processing unit, operably coupled to the memory, to perform, during a first computational cycle, at least one calculation on the first portion of the input data so as to generate first output data; and

a controller, operably coupled to the processing unit and the memory, to store the first output data in a second partition of the memory, the second partition referenced by a second pointer, and to swap the first pointer with the second pointer at the end of the first computational cycle, such that the first output data becomes an input for a second computational cycle of the numerical simulation.

113. (New) The system of claim 112, wherein the processing unit comprises a graphics processing unit.

114. (New) The system of claim 112, wherein the controller is configured to send instructions for performing the at least one calculation to the processing unit.

115. (New) The system of claim 112, wherein the accelerator controller is further configured to transfer the first output data to another memory during the second computational cycle.

116. (New) The system of claim 112, wherein the memory further comprises:

a third partition to store internal variables; and

a fourth partition to store data used as input at a particular computation cycle of the numerical simulation.

117. (New) A computer system, comprising:

a central processing unit to receive input data;

main memory, operably coupled to the central processing unit via a bus, to store the input data received by the central processing unit;

an accelerator, operably coupled to the central processing unit and the main memory via the bus, to receive at least a portion of the input data from the main memory, the accelerator comprising:

at least one processing unit to perform a sequence of computations on the at least a portion of the input data so as to generate output data, intermediate computations in the sequence of computations yielding intermediate results; and

accelerator memory, operably coupled to the at least one processing unit, to store the results of the sequence of computations; and

a controller, operably coupled to the at least one processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory and to transfer at least a portion of the output data from the accelerator memory to the main memory during performance of the sequence of computations by the at least one graphic processing unit.

118. (New) The computer system of claim 117, wherein the central processing unit is configured to receive the input data in response to a user interaction.

119. (New) The computer system of claim 117, wherein:

the central processing unit is configured to receive the input data at a first rate; and

the at least one processing unit is configured to perform the sequence of computations at a second rate different than the first rate.

120. (New) The computer system of claim 117, wherein the main memory is configured to store a copy of the output data stored in the accelerator memory.

121. (New) The computer system of claim 117, wherein an output of at least one computation in the sequence of computations represents an output of at least one neuron in an artificial neural network.

122. (New) The computer system of claim 117, wherein accelerator memory comprises:

a first memory partition to store parameters common to all of the computations in the sequence of computations; and

a second memory partition to store data specific to at least one computation in the sequence of computations.

123. (New) The computer system of claim 117, wherein the controller is configured to transfer the output data from the accelerator memory to the main memory without transferring any of the intermediate results from the accelerator memory to the main memory so as to reduce data transfer via the bus.

124. (New) The computer system of claim 117, wherein the controller is configured to transfer at least a portion of the output data from the accelerator memory to the main memory after the at least one processing unit has begun to perform another sequence of computations.

125. (New) The computer system of claim 124, wherein the controller is configured to initiate transfer of the at least a portion of the input data and to transfer the at least a portion of the output data in parallel with performance of at least one computation in the other sequence of computations by the at least one processing unit.

126. (New) The computer system of claim 117, wherein the controller is configured to control execution of the sequence of computations by the at least one processing unit.

127. (New) The computer system of claim 117, further comprising:

at least one of a video camera, a microphone, or a cell recording electrode, operably coupled to the central processor unit, to acquire the input data in real time.

128. (New) A method of performing a sequence of computations on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a processing unit and an accelerator memory, the method comprising:

(A) performing, by the processing unit, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data, intermediate computations in the sequence of computations yielding intermediate results;

(B) in parallel with performing the sequence of computations by the processing unit in (A), transferring a second portion of the input data from the main memory to the accelerator via the bus; and

(C) in parallel with performing the sequence of computations by the processing unit in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus.

129. (New) The method of claim 128, further comprising:

storing the input data in the main memory in response to a user interaction.

130. (New) The method of claim 128, further comprising:

receiving the input data at a first rate; and

wherein (A) comprises performing the sequence of computations at a second rate different than the first rate.

131. (New) The method of claim 128, wherein (A) comprises:

generating an output representative of an output of at least one neuron in an artificial neural network.

132. (New) The method of claim 128, wherein (C) comprises:

transferring the second portion of the output data from the accelerator memory to the main memory without transferring any of the intermediate results of the plurality of sequential computations from the accelerator memory to the main memory so as to reduce data transfer via the bus.

133. (New) The method of claim 128, wherein (C) comprises:

transferring the second portion of the output data from the accelerator memory to the main memory after the processing unit has begun to perform another sequence of computations.

134. (New) The method of claim 133, wherein (C) further comprises:

initiating transfer of the second portion of the output data in parallel with performance of at least one computation in the other sequence of computations.

135. (New) The method of claim 128, further comprising:

acquiring the input data in real time with at least one of a video camera, a microphone, or a cell recording electrod operably coupled to the processing unit.

136. (New) The method of claim 128, further comprising:

storing parameters common to all of the computations in the sequence of computations in a first memory partition in the accelerator memory; and

storing data specific to at least one computation in the sequence of computations in a second memory partition in the accelerator memory.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	Gorchetchnikov *et al.*	Confirmation No.:	Not Yet Assigned
Serial No.:	Not Yet Assigned	Group Art Unit:	Not Yet Assigned
Filed:	Herewith	Examiner:	Not Yet Assigned

FOR: **GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD**

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PRELIMINARY AMENDMENT

Prior to examination of the above-identified application, please amend the above-captioned application as follows:

Amendments to the Claims are reflected on the listing of the claims which begins on page 2 of this paper.

Remarks begin on page 23 of this paper.

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 15/808,201	Filing Date 11/09/2017	☐ To be Mailed

ENTITY: ☐ LARGE ☒ SMALL ☐ MICRO

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	11/09/2017	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 136	Minus	** 136	= 0	X $40 =	0
	Independent (37 CFR 1.16(h))	* 15	Minus	***15	= 0	X $210 =	0
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))						
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))						
						TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

LIE
PATRICIA F. LEWIS

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Electronic Acknowledgement Receipt

EFS ID:	31169504
Application Number:	15808201
International Application Number:	
Confirmation Number:	1077
Title of Invention:	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD
First Named Inventor/Applicant Name:	Anatoli Gorchetchnikov
Customer Number:	58249
Filer:	Nisha Somnath/Donna Doyle
Filer Authorized By:	Nisha Somnath
Attorney Docket Number:	NRLA-001/03US 320730-2040
Receipt Date:	08-DEC-2017
Filing Date:	
Time Stamp:	19:57:06
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Non Patent Literature	Versace_FromSpikestoInter-ArealSynchrony.pdf	21460 d5cf282a078982ffef9358c8217f35090d530395	no	1

Warnings:

Information:					
2	Non Patent Literature	Verasace_Open_Source_Computer_Software_Engineering_Applications_and_Trend_Versace.pdf	1096267 3b4a1caf72bea01b17ae4977508b5e451d909888	no	43
Warnings:					
Information:					
3	Non Patent Literature	Versace_KInNeSSAmodularframeworkforcomputationalneuroscience.pdf	836843 bbce576553e7642893603d73cabe99099141276b	no	31
Warnings:					
Information:					
4	Non Patent Literature	Versace_MoNETAAMindMadefromMemristors.pdf	5120927 b5cc3438eb44a0d08586e677b6b132ab6cfe1478	no	8
Warnings:					
Information:					
5	Non Patent Literature	Webster.pdf	8621811 973ccb15e5b617ff05b9dd247d3db855ac22bcf6	no	14
Warnings:					
Information:					
6	Non Patent Literature	Wiskott_SlowfeatureanalysisUnsupervisedlearningofinvariances.pdf	1327221 6ebab1e6f1010667781ee97772dc4c79bf874bde	no	56
Warnings:					
Information:					
7	Non Patent Literature	Gorchetchnikov_Anapproach.pdf	3940577 deec9e7fd055ad0d08339c8be193d6096ecbfcda	no	70
Warnings:					
Information:					
8	Non Patent Literature	Versace_TedX_Fulbright_2014.pdf	4455341 aba040ae40d2f5ecd2790566960f76728498d775	no	30
Warnings:					
Information:					
		Total Files Size (in bytes):	25420447		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

NRLA-00601WO
320150-2020
C. Colice

PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

To: JOSEPH TEJA, JR.
COOLEY LLP
1299 PENNSYLVANIA AVENUE, NW
SUITE 700
WASHINGTON, DC 20004-2400

PCT

NOTIFICATION OF TRANSMITTAL OF THE INTERNATIONAL SEARCH REPORT AND THE WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY, OR THE DECLARATION

(PCT Rule 44.1)

	Date of mailing *(day/month/year)* 23 FEB 2016
Applicant's or agent's file reference NRLA00601WO	**FOR FURTHER ACTION** See paragraphs 1 and 4 below
International application No. PCT/US 15/29438	International filing date *(day/month/year)* 06 May 2015 (06.05.2015)
Applicant NEURALA, INC.	

DOCKETED
DUE Art. 19 Amendments
DATE
DEADLINE 4/23/16

1. ☒ The applicant is hereby notified that the international search report and the written opinion of the International Searching Authority have been established and are transmitted herewith.

   Filing of amendments and statement under Article 19:
   The applicant is entitled, if he so wishes, to amend the claims of the international application (see Rule 46):

   When? The time limit for filing such amendments is normally two months from the date of transmittal of the international search report.

   How? Directly to the International Bureau of WIPO preferably through ePCT or on paper to, 34 chemin des Colombettes 1211 Geneva 20, Switzerland, Facsimile No.: +41 22 338 82 70

   For more detailed instructions, see *PCT Applicant's Guide*, International Phase, paragraphs 9.004 – 9.011.

2. ☐ The applicant is hereby notified that no international search report will be established and that the declaration under Article 17(2)(a) to that effect and the written opinion of the International Searching Authority are transmitted herewith.

3. ☐ **With regard to any protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:
   ☐ the protest together with the decision thereon has been transmitted to the International Bureau together with any request to forward the texts of both the protest and the decision thereon to the designated Offices.
   ☐ no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Reminders**

   The applicant may **submit comments on an informal basis on the written opinion of the International Searching Authority** to the International Bureau. These comments will be made available to the public after international publication. The International Bureau will send a copy of such comments to all designated Offices unless an international preliminary examination report has been or is to be established.

   Shortly after the expiration of **18 months from the priority date, the international application will be published** by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau before the completion of the technical preparations for international publication (Rules 90*bis*.1 and 90*bis*.3).

   Within **19 months** from the priority date, but only in respect of some designated Offices, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase **until 30 months** from the priority date (in some Offices even later); otherwise, the applicant must, **within 20 months** from the priority date, perform the prescribed acts for **entry into the national phase** before those designated Offices. In respect of other designated Offices, the time limit of **30 months** (or later) will apply even if no demand is filed within 19 months. For details about the applicable time limits, Office by Office, see www.wipo.int/pct/en/texts/time_limits.html and the *PCT Applicant's Guide*, National Chapters.

   Within **19 months from the priority date, the applicant may request that a supplementary international search be carried out** by a different International Searching Authority that offers this service (Rule 45*bis*.1). The procedure for requesting supplementary international search is described in the *PCT Applicant's Guide*, International Phase, paragraphs 8.006-8.032.

Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Lee W. Young PCT Helpdesk: 571-272-4300 Telephone No. PCT OSP: 571-272-7774

Received
FEB 26 2016

Form PCT/ISA/220 (July 2014)

Cooley LLP

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference NRLA00601WO	**FOR FURTHER ACTION**	see Form PCT/ISA/220 as well as, where applicable, item 5 below.
International application No. PCT/US 15/29438	International filing date *(day/month/year)* 06 May 2015 (06.05.2015)	(Earliest) Priority Date *(day/month/year)* 06 May 2014 (06.05.2014)
Applicant NEURALA, INC.		

This international search report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This international search report consists of a total of 4 sheets.

☐ It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**

   a. With regard to the **language**, the international search was carried out on the basis of:

   ☒ the international application in the language in which it was filed.

   ☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

   b. ☐ This international search report has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43.6*bis*(a)).

   c. ☐ With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, see Box No. I.

2. ☐ **Certain claims were found unsearchable** (see Box No. II).

3. ☒ **Unity of invention is lacking** (see Box No. III).

4. With regard to the **title**,

   ☒ the text is approved as submitted by the applicant.

   ☐ the text has been established by this Authority to read as follows:

5. With regard to the **abstract**,

   ☒ the text is approved as submitted by the applicant.

   ☐ the text has been established, according to Rule 38.2, by this Authority as it appears in Box No. IV. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. With regard to the **drawings**,

   a. the figure of the **drawings** to be published with the abstract is Figure No. 11A

   ☒ as suggested by the applicant.

   ☐ as selected by this Authority, because the applicant failed to suggest a figure.

   ☐ as selected by this Authority, because this figure better characterizes the invention.

   b. ☐ none of the figures is to be published with the abstract.

NVIDIA Ex1002 Page 76 of 1081

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 15/29438

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fees must be paid.

Group I: Claims 1-13 and 17-19 directed to a method and system for generating a hardware-agnostic behavior of at least one electronic device.

Group II: Claims 14-16 directed to a method for loading at least one hardware-agnostic behavior between a first robot and a second robot.

Group III: Claim 20 directed to a method of engaging at least one hardware-agnostic behavior to control at least one robot.

--- (See Continuation in Supplemental Box) ---

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:
1-13 and 17-19

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 15/29438

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06N 3/08 (2016.01)
CPC - G06N 3/08
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC (8) - G06N 3/08 (2016.01)
CPC - G06N 3/08

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
CPC - G06N 3/10, G06N 3/049 (See Keywords Below)
USPC - 706/25, 706/44, 706/11

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Thomsoninnovation.com; Patbase; Google Scholar; Google Patents; Gogole.com; Freepatentsonline; ProQuest Dialog
Search Terms: Robot, brain, define, program, neuron, behavior, dendrite, trigger, stimuli, response, action, axon, download, transfer, convert, translate, hardware independent, agnostic, network, interface, GUI, user interface, etc.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2014/0052679 A1 (SINYAVSKIY et al.), 20 February 2014 (20.02.2014), entire document, especially Abstract; Para [0083], [0088], [0094]-[0097], [0100], [0355]-[0358]	1-13 and 17-19
Y	US 2014/0089232 A1 (BUIBAS et al.), 27 March 2014 (27.03.2014), entire document, especially Abstract; Para [0016], [0021], [0120], [0183]-[0185], [0234]-[0237]	1-13 and 17-19
A	US 2008/0066065 A1 (KIM et al.), 13 March 2008 (13.03.2008), entire document	1-13 and 17-19
A	US 2006/0184273 A1 (SAWADA et al.), 17 August 2006 (17.08.2006), entire document	1-13 and 17-19
A	US 2014/0019392 A1 (BUIBAS), 16 January 2014 (16.01.2014), entire document	1-13 and 17-19

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
08 February 2016 (08.02.2016)	23 FEB 2016
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 15/29438

Continuation of:
Box III. Observations where unity of invention is lacking

The inventions listed as Groups I-III do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons:

Special Technical Features:

The invention of Group I included the features of receiving, via a user interface, at least one stimulus selection from a user, the at least one stimulus selection corresponding to at least one stimulus detectable by the at least one electronic device; receiving, via the user interface, at least one hardware-agnostic response selection from the user, the at least one hardware-agnostic response selection corresponding to at least one action to be performed by the at least one electronic device in response to the at least one stimulus; and generating, via a processor operably coupled to the user interface, the hardware-agnostic behavior based on the at least one stimulus selection received in A) and the at least one hardware-agnostic response selection received in B); rendering, via the user interface, the behavior as a behavior neuron comprising at least one dendrite representing the at least one stimulus and at least a portion of a neuron axon representing the at least one response, not required in any other groups.

The invention of Group II included the features of receiving a request to load a first hardware-agnostic behavior onto the first robot; retrieving the first hardware-agnostic behavior from at least one storage device, the first hardware-agnostic behavior defining at least one first hardware-agnostic robot response to at least one first hardware-agnostic robot sensor stimulus; providing the first hardware-agnostic behavior to the first robot; providing the first hardware-agnostic behavior to the second robot; receiving a request to load a second hardware-agnostic behavior onto the first robot, the second hardware-agnostic behavior defining at least one second hardware-agnostic robot response to at least one second hardware-agnostic robot sensor stimulus; retrieving the second hardware-agnostic behavior from the at least one storage device; providing the second hardware-agnostic behavior to the first robot, not required in any other groups.

The invention of Group III included the features of establishing a communications connection between the at least one robot and a graphical user interface (GUI); receiving, via the GUI, an indication from a user regarding selection of the at least one hardware-agnostic behavior; retrieving, from a memory operably coupled to the GUI, instructions for causing the at least one robot to operate according to the at least one hardware-agnostic behavior; and executing the instructions with a processor operably coupled to the GUI so as to engage the at least one hardware-agnostic behavior to control the at least one robot, not required in any other groups.

Common Technical Features

Groups I-III share the technical features of the hardware-agnostic behavior

Groups I-II share the technical features of stimulus detectable by the at least one electronic device; hardware-agnostic robot response.

Groups II-III share the technical features of a robot.

However, the shared technical features does not represent a contribution over prior art as anticipated by US 2006/0184273 A1 to Sawada et al. (hereinafter 'Sawada'), 17 August 2006 (17.08.2016).

Sawada teaches the features of a hardware-agnostic behavior (para [0203], [0207]- reflexive situated behaviors layer 103 is a functional module to implement a reflexive situated behavior in response to an external stimulus recognized by the video recognition unit 81, audio recognition unit 82 or tactile recognition unit 83; situated behaviors layer (SBL) 102 and reflexive situated behaviors layer 103 can be stated as superordinate application programs independent of the hardware of the robot device 1); stimulus detectable by the at least one electronic device (para [0080]- external-stimulus recognition unit 80 to recognize an external stimulus); hardware-agnostic robot response (para [0078], [0155], [0207] - the robot device 1 can autonomously select and implement an action in response to an external stimulus and its own internal state; actions in response to the same external stimulus; application programs independent of the hardware of the robot device); a robot (Abstract - robot device).

As the common features were known in the art at the time of the invention, this cannot be considered a common technical feature that would otherwise unify the groups.

Therefore, Groups I-III lack unity under PCT Rule 13.

PATENT COOPERATION TREATY

From the
INTERNATIONAL SEARCHING AUTHORITY

To: JOSEPH TEJA, JR.
COOLEY LLP
1299 PENNSYLVANIA AVENUE, NW
SUITE 700
WASHINGTON, DC 20004-2400

PCT

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

(PCT Rule 43*bis*.1)

Date of mailing *(day/month/year)*	23 FEB 2016
Applicant's or agent's file reference NRLA00601WO	**FOR FURTHER ACTION** See paragraph 2 below

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/US 15/29438	06 May 2015 (06.05.2015)	06 May 2014 (06.05.2014)

International Patent Classification (IPC) or both national classification and IPC
IPC(8) - G06N 3/08 (2016.01)
CPC - G06N 3/08

Applicant NEURALA, INC.

1. This opinion contains indications relating to the following items:

DOCKETED Smm
DUE Req. to Written
DATE Opinion
DEADLINE 5/23/16

- [x] Box No. I Basis of the opinion
- [ ] Box No. II Priority
- [ ] Box No. III Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- [x] Box No. IV Lack of unity of invention
- [x] Box No. V Reasoned statement under Rule *43bis*.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- [ ] Box No. VI Certain documents cited
- [ ] Box No. VII Certain defects in the international application
- [ ] Box No. VIII Certain observations on the international application

2. **FURTHER ACTION**

If a demand for international preliminary examination is made, this opinion will be considered to be a written opinion of the International Preliminary Examining Authority ("IPEA") except that this does not apply where the applicant chooses an Authority other than this one to be the IPEA and the chosen IPEA has notified the International Bureau under Rule 66.1 *bis*(b) that written opinions of this International Searching Authority will not be so considered.

If this opinion is, as provided above, considered to be a written opinion of the IPEA, the applicant is invited to submit to the IPEA a written reply together, where appropriate, with amendments, before the expiration of 3 months from the date of mailing of Form PCT/ISA/220 or before the expiration of 22 months from the priority date, whichever expires later.

For further options, see Form PCT/ISA/220.

Name and mailing address of the ISA/US	Date of completion of this opinion	Authorized officer:
Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	08 February 2016 (08.02.2016)	Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/237 (cover sheet) (January 2015)

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 15/29438

Box No. I Basis of this opinion

1. With regard to the **language**, this opinion has been established on the basis of:

   ☒ the international application in the language in which it was filed.

   ☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

2. ☐ This opinion has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43*bis*.1(a)).

3. ☐ With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, this opinion has been established on the basis of a sequence listing:

   a. ☐ forming part of the international application as filed:

      ☐ in the form of an Annex C/ST.25 text file.

      ☐ on paper or in the form of an image file.

   b. ☐ furnished together with the international application under PCT Rule 13*ter*.1(a) for the purposes of international search only in the form of an Annex C/ST.25 text file.

   c. ☐ furnished subsequent to the international filing date for the purposes of international search only:

      ☐ in the form of an Annex C/ST.25 text file (Rule 13*ter*.1(a)).

      ☐ on paper or in the form of an image file (Rule 13*ter*.1(b) and Administrative Instructions, Section 713).

4. ☐ In addition, in the case that more than one version or copy of a sequence listing has been filed or furnished, the required statements that the information in the subsequent or additional copies is identical to that forming part of the application as filed or does not go beyond the application as filed, as appropriate, were furnished.

5. Additional comments:

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 15/29438

Box No. IV Lack of unity of invention

1. ☒ In response to the invitation (Form PCT/ISA/206) to pay additional fees the applicant has, within the applicable time limit:

   ☐ paid additional fees.

   ☐ paid additional fees under protest and, where applicable, the protest fee.

   ☐ paid additional fees under protest but the applicable protest fee was not paid.

   ☒ not paid additional fees.

2. ☐ This Authority found that the requirement of unity of invention is not complied with and chose not to invite the applicant to pay additional fees.

3. This Authority considers that the requirement of unity of invention in accordance with Rule 13.1, 13.2 and 13.3 is

   ☐ complied with.

   ☒ not complied with for the following reasons:

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fees must be paid.

Group I: Claims 1-13 and 17-19 directed to a method and system for generating a hardware-agnostic behavior of at least one electronic device.

Group II: Claims 14-16 directed to a method for loading at least one hardware-agnostic behavior between a first robot and a second robot.

Group III: Claim 20 directed to a method of engaging at least one hardware-agnostic behavior to control at least one robot.

The inventions listed as Groups I-III do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons:

Special Technical Features:

The invention of Group I included the features of receiving, via a user interface, at least one stimulus selection from a user, the at least one stimulus selection corresponding to at least one stimulus detectable by the at least one electronic device; receiving, via the user interface, at least one hardware-agnostic response selection from the user, the at least one hardware-agnostic response selection corresponding to at least one action to be performed by the at least one electronic device in response to the at least one stimulus; and generating, via a processor operably coupled to the user interface, the hardware-agnostic behavior based on the at least one stimulus selection received in A) and the at least one hardware-agnostic response selection received in B); rendering, via the user interface, the behavior as a behavior neuron comprising at least one dendrite representing the at least one stimulus and at least a portion of a neuron axon representing the at least one response, not required in any other groups.

The invention of Group II included the features of receiving a request to load a first hardware-agnostic behavior onto the first robot; retrieving the first hardware-agnostic behavior from at least one storage device, the first hardware-agnostic behavior defining at least one first hardware-agnostic robot response to at least one first hardware-agnostic robot sensor stimulus; providing the first hardware-agnostic behavior to the first robot; providing the first hardware-agnostic behavior to the second robot; receiving a request to load a second hardware-agnostic behavior onto the first robot, the second hardware-agnostic behavior defining at least one second hardware-agnostic robot response to at least one second hardware-agnostic robot sensor stimulus; retrieving the second hardware-agnostic behavior from the at least one storage device; providing the second hardware-agnostic behavior to the first robot, not required in any other groups.

The invention of Group III included the features of establishing a communications connection between the at least one robot and a graphical user interface (GUI); receiving, via the GUI, an indication from a user regarding selection of the at least one hardware-agnostic behavior; retrieving, from a memory operably coupled to the GUI, instructions for causing the at least one robot to operate according to the at least one hardware-agnostic behavior; and executing the instructions with a processor operably coupled to the GUI so as to engage the at least one hardware-agnostic behavior to control the at least one robot, not required in any other groups.

--- (See Continuation in Supplemental Box) ---

4. Consequently, this opinion has been established in respect of the following parts of the international application:

   ☐ all parts.

   ☒ the parts relating to claims Nos. 1-13 and 17-19

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 15/29438

Box No. V **Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement**

1. Statement

Novelty (N)	Claims	1-13 and 17-19	YES
	Claims	None	NO
Inventive step (IS)	Claims	None	YES
	Claims	1-13 and 17-19	NO
Industrial applicability (IA)	Claims	1-13 and 17-19	YES
	Claims	None	NO

2. Citations and explanations:

Claims 1-13 and 17-19 and lack an inventive step under PCT Article 33(3) as being obvious over US 2014/0052679 A1 to Sinyavskiy et al. (hereinafter 'Sinyavskiy') in view of US 2014/0089232 A1 to Buibas et al. (hereinafter 'Buibas').

Regarding claim 1, Sinyavskiy teaches a method for generating a hardware-agnostic behavior of at least one electronic device, the method comprising (Abstract): receiving at least one stimulus selection corresponding to at least one stimulus detectable by the at least one electronic device (Para [0013]-[0014], [0087] the input connection may be configured to provide a feed-forward stimulus to the neuron; the feed-forward stimulus may comprise a plurality of spikes characterized by an inter spike interval (ISI) comprising an interval between individual of the plurality of spikes; a dendritic (or an internal) event may be defined in order, for example, to modify input connections of a neuron. The internal event may be triggered by an external signal and/or internally by the neuron); receiving at least one hardware-agnostic response corresponding to at least one action to be performed by the at least one electronic device in response to the at least one stimulus (Para [0083], [0088], [0094]-[0097], [0100]- an axonic (or delivery) event may be defined in order, for example, to cause distribution of post-synaptic response of a neuron to appropriate destinations; the axonic event may be triggered by the neuron spike response process (e.g., the super-threshold condition); the instruction set--Elementary Network Description (END) format--that is optimized for efficient representation of neuronal systems in hardware-independent manner.; the END description is hardware-agnostic); generating, via a processor, the hardware agnostic behavior based on the at least one stimulus and the at least one hardware-agnostic response (Para [0093], [0355]-[0356]- the dendritic and the axonic events may be combined to obtain a composite event, such as, for example, the post-synaptic spike used, for example, in used in the elementary neuromorphic description (END) framework v 1, v2, and v3; the external reinforcement signal 1408 may be used to trigger plasticity updates of one or more input connections 1404 (e.g., modify weights 1406) based on the neuron prior performance, in one or more implementations). Sinyavskiy does not teach a user interface for receiving user selections. However, Buibas teaches a user interface for receiving user selections (Abstract; Para [0016], [0021] - browsing a plurality of state files via a user interface, the state files being stored on the cloud server apparatus; selecting an individual one of the plurality of state files). It would have been obvious to one of ordinary skill in the art to combine teachings of Sinyavskiy with those of Buibas since doing so would provide a user friendly and interactive graphical system for programming and visualizing behavior of robotic systems.

Regarding claim 2, Sinyavskiy teaches the at least one electronic device comprises at least one robot (Para [0161]).

Regarding claim 3, Sinyavskiy teaches the at least one stimulus is based at least in part on an output from a neural network (Para [0033], [0159]-[0161], [0164], [0374]-[0377]).

Regarding claim 4, Sinyavskiy teaches the at least one stimulus comprises sensing at least one of: depressing a button; swiping a touchscreen; a change in attitude with a gyroscope; acceleration with an accelerometer; a change in battery charge; a wireless signal strength; a time of day; a date; passage of a predetermined time period; magnetic field strength; electric field strength; stress; strain; position; altitude; speed; velocity; angular velocity; trajectory; a face, object, and/or scene with an imaging detector; motion; touch; and sound and/or speech with a microphone (Para [0165]-[0166] - accelerometer).

Regarding claim 5, Sinyavskiy teaches the at least one response is based at least in part on an output from a neural network (Para [0095]-[0097], [0407]-[0410], [0413]).

Regarding claim 6, Sinyavskiy teaches the output from the neural network comprises at least one of a visual object or an auditory object recognized by the neural network (Para [0413]-[0415]).

Regarding claim 7, Buibas teaches receiving, via the user interface, a selection of at least one particular electronic device to associate with the hardware-agnostic behavior generated and associating the hardware-agnostic behavior generated with the at least one particular electronic device selected (Para [0120], [0183]-[0185], [0234]-[0237]).

Regarding claim 8, Buibas teaches determining identifying information for the at least one particular electronic device, the identifying information including information about at least one sensor and/or at least one actuator associated with the at least one particular electronic device (Para [0102]-[0103], [0183]-[0185]).

--- (See Continuation in Supplemental Box) ---

NVIDIA Ex1002 Page 83 of 1081

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.
Continuation of:
Box IV. Lack of unity of invention

Common Technical Features

Groups I-III share the technical features of the hardware-agnostic behavior

Groups I-II share the technical features of stimulus detectable by the at least one electronic device; hardware-agnostic robot response.

Groups II-III share the technical features of a robot.

However, the shared technical features does not represent a contribution over prior art as anticipated by US 2006/0184273 A1 to Sawada et al. (hereinafter 'Sawada'), 17 August 2006 (17.08.2016).

Sawada teaches the features of a hardware-agnostic behavior (para [0203], [0207]- reflexive situated behaviors layer 103 is a functional module to implement a reflexive situated behavior in response to an external stimulus recognized by the video recognition unit 81, audio recognition unit 82 or tactile recognition unit 83; situated behaviors layer (SBL) 102 and reflexive situated behaviors layer 103 can be stated as superordinate application programs independent of the hardware of the robot device 1); stimulus detectable by the at least one electronic device (para [0080]- external-stimulus recognition unit 80 to recognize an external stimulus); hardware-agnostic robot response (para [0078], [0155], [0207] - the robot device 1 can autonomously select and implement an action in response to an external stimulus and its own internal state; actions in response to the same external stimulus; application programs independent of the hardware of the robot device); a robot (Abstract - robot device).

As the common features were known in the art at the time of the invention, this cannot be considered a common technical feature that would otherwise unify the groups.

Therefore, Groups I-III lack unity under PCT Rule 13.

Box V.2. Citations and Explanations.

Regarding claim 9, Buibas teaches translating the hardware-agnostic behavior into hardware-specific instructions based at least in part on the identifying information and providing the hardware-specific instructions to the at least one particular electronic device (Para [0088]-[0089], [0101]-[0102], [0116]-[0120]).

Regarding claim 10, Buibas teaches the at least one hardware-agnostic response selection comprises a sequence of actions to be performed by the at least one electronic device in response to at least one corresponding stimulus (Para [0086]-[0088], [0179], [0212]).

Regarding claim 11, Sinyavskiy teaches generating, via the processor, at least one other hardware-agnostic behavior based on at least one other stimulus selection and at least one other hardware-agnostic response selection and forming a hardware-agnostic personality based at least on the hardware-agnostic robot behavior and at least one other hardware-agnostic robot behavior (Para [0098]-[0100], [0159]-[0163]).

Regarding claim 12, Sinyavskiy teaches a system for generating a hardware-agnostic behavior of at least one electronic device, the system comprising: receiving at least one stimulus selection corresponding to at least one stimulus detectable by the at least one electronic device (Para [0013]-[0014], [0087] the input connection may be configured to provide a feed-forward stimulus to the neuron; the feed-forward stimulus may comprise a plurality of spikes characterized by an inter spike interval (ISI) comprising an interval between individual of the plurality of spikes; a dendritic (or an internal) event may be defined in order, for example, to modify input connections of a neuron. The internal event may be triggered by an external signal and/or internally by the neuron); receiving at least one hardware-agnostic response corresponding to at least one action to be performed by the at least one electronic device in response to the at least one stimulus (Para [0083], [0088], [0094]-[0097], [0100]- an axonic (or delivery) event may be defined in order, for example, to cause distribution of post-synaptic response of a neuron to appropriate destinations; the axonic event may be triggered by the neuron spike response process (e.g., the super-threshold condition); the instruction set--Elementary Network Description (END) format--that is optimized for efficient representation of neuronal systems in hardware-independent manner.; the END description is hardware-agnostic); generating, via a processor, the hardware agnostic behavior based on the at least one stimulus and the at least one hardware-agnostic response (Para [0093], [0355]-[0358]- the dendritic and the axonic events may be combined to obtain a composite event, such as, for example, the post-synaptic spike used, for example, in used in the elementary neuromorphic description (END) framework v 1, v2, and v3; the external reinforcement signal 1408 may be used to trigger plasticity updates of one or more input connections 1404 (e.g., modify weights 1406) based on the neuron prior performance, in one or more implementations). Sinyavskiy does not teach a user interface for receiving user selections; communications port, operably coupled to the processor, to provide the hardware agnostic behavior to the at least one electronic device. However, Buibas teaches a user interface for receiving user selections (Abstract; Para [0016], [0021] - browsing a plurality of state files via a user interface, the state files being stored on the cloud server apparatus; selecting an individual one of the plurality of state files); communications port, operably coupled to the processor, to provide the hardware agnostic behavior to the at least one electronic device (Para [0090]-[0093], [0122]-[0123], [0183]- operating several identical robotic devices and the network weight configuration of a single 'trained' apparatus is downloaded into other (untrained) devices). It would have been obvious to one of ordinary skill in the art to combine teachings of Sinyavskiy with those of Buibas since doing so would provide a user friendly and interactive graphical system for programming and visualizing behavior of robotic systems.

--- (See Continuation in Supplemental Box) ---

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.

Continuation of:
Box V.2. Citations and Explanations.

Regarding claim 13, Buibas teaches a hardware translation component, operably coupled to the communications port, to translate the hardware-agnostic behavior into a set of hardware-specific input triggers to be sensed by the at least one electronic device and a set of hardware-specific actions in response to the set of hardware-specific input triggers to be performed by the at least one electronic device (Para [0088]-[0089], [0101]-[0102], [0116]-[0120]).

Regarding claim 17, Sinyavskiy teaches a computer-implemented method for generating behaviors for a robot, the method comprising (Abstract): Receiving at least one stimulus to be sensed by the robot (Para [0013]-[0014], [0087], [0161]- the input connection may be configured to provide a feed-forward stimulus to the neuron; the feed-forward stimulus may comprise a plurality of spikes characterized by an inter spike interval (ISI) comprising an interval between individual of the plurality of spikes; a dendritic (or an internal) event may be defined in order, for example, to modify input connections of a neuron. The internal event may be triggered by an external signal and/or internally by the neuron; the motor interface may comprise a motor interface configured to control an external robotic apparatus); Receiving at least one response to be performed by the robot (Para [0083], [0088], [0094]-[0097], [0100]- an axonic (or delivery) event may be defined in order, for example, to cause distribution of post-synaptic response of a neuron to appropriate destinations; the axonic event may be triggered by the neuron spike response process (e.g., the super-threshold condition); the instruction set--Elementary Network Description (END) format--that is optimized for efficient representation of neuronal systems in hardware-independent manner.; the END description is hardware-agnostic); generating a behavior for the robot based at least in part on the at least one stimulus and the at least one response (Para [0093], [0355]-[0358]- the dendritic and the axonic events may be combined to obtain a composite event, such as, for example, the post-synaptic spike used, for example, in used in the elementary neuromorphic description (END) framework v 1, v2, and v3; the external reinforcement signal 1408 may be used to trigger plasticity updates of one or more input connections 1404 (e.g., modify weights 1406) based on the neuron prior performance, in one or more implementations); rendering the behavior as a behavior neuron comprising at least one dendrite representing the at least one stimulus and at least a portion of a neuron axon representing the at least one response (para [0087]-[0088], [0093]-[0097], [0099]- The computational engine is capable of saving its current state in the same format for subsequent re-uploading. The binary data is used by the monitoring and visualization block 110 of FIG. 1 during debugging and visualization of computation results and monitoring of the system; axonic (or delivery) event may be defined in order, for example, to cause distribution of post-synaptic response of a neuron; a dendritic (or an internal) event may be defined in order, for example, to modify input connections of a neuron). Sinyavskiy does not teach a user interface for receiving user selections. However, Buibas teaches a user interface for receiving user selections (Abstract; Para [0016], [0021] - browsing a plurality of state files via a user interface, the state files being stored on the cloud server apparatus; selecting an individual one of the plurality of state files). It would have been obvious to one of ordinary skill in the art to combine teachings of Sinyavskiy with those of Buibas since doing so would provide a user friendly and interactive graphical system for programming and visualizing behavior of robotic systems.

Regarding claim 18, Sinyavskiy teaches rendering the behavior neuron as at least one neuron of a plurality of neurons in a graphical representation of a brain (para [0087]-[0088], [0093]-[0097], [0099]).

Regarding claim 19, Sinyavskiy teaches rendering the at least one neuron in the graphical representation of the brain based on the nature of the behavior in relation to behavior centers of an animal brain (para [0087]-[0088], [0093]-[0097], [0099]).

Claims 1-13 and 17-19 have industrial applicability as defined by PCT Article 33(4) because the subject matter can be made or used in industry.

C. COLICE
320730-2036

PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

To: CHRISTOPHER MAX COLICE
COOLEY LLP
1299 PENNSYLVANIA AVENUE, NW SUITE 700
WASHINGTON, DC 20004-2400

PCT

NOTIFICATION OF TRANSMITTAL OF THE INTERNATIONAL SEARCH REPORT AND THE WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY, OR THE DECLARATION

(PCT Rule 44.1)

Applicant's or agent's file reference NRLA00901WO	Date of mailing *(day/month/year)* **FOR FURTHER ACTION** See paragraphs 1 and 4 below
International application No. PCT/US2017/029866	International filing date *(day/month/year)* 27 April 2017
Applicant NEURALA, INC.	

1. ☒ The applicant is hereby notified that the international search report and the written opinion of the International Searching Authority have been established and are transmitted herewith.

   Filing of amendments and statement under Article 19:
   The applicant is entitled, if he so wishes, to amend the claims of the international application (see Rule 46):

   When? The time limit for filing such amendments is normally two months from the date of transmittal of the international search report.

   How? Directly to the International Bureau of WIPO preferably through ePCT or on paper to, 34 chemin des Colombettes 1211 Geneva 20, Switzerland, Facsimile No.: +41 22 338 82 70

   For more detailed instructions, see *PCT Applicant's Guide*, International Phase, paragraphs 9.004 – 9.011.

2. ☐ The applicant is hereby notified that no international search report will be established and that the declaration under Article 17(2)(a) to that effect and the written opinion of the International Searching Authority are transmitted herewith.

3. ☐ **With regard to any protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:
   ☐ the protest together with the decision thereon has been transmitted to the International Bureau together with any request to forward the texts of both the protest and the decision thereon to the designated Offices.
   ☐ no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Reminders**

   The applicant may **submit comments on an informal basis on the written opinion of the International Searching Authority** to the International Bureau. These comments will be made available to the public after international publication. The International Bureau will send a copy of such comments to all designated Offices unless an international preliminary examination report has been or is to be established.

   Shortly after the expiration of **18 months from the priority date, the international application will be published** by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau before the completion of the technical preparations for international publication (Rules 90*bis*.1 and 90*bis*.3).

   Within **19 months** from the priority date, but only in respect of some designated Offices, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase **until 30 months** from the priority date (in some Offices even later); otherwise, the applicant must, **within 20 months** from the priority date, perform the prescribed acts for **entry into the national phase** before those designated Offices. In respect of other designated Offices, the time limit of **30 months** (or later) will apply even if no demand is filed within 19 months. For details about the applicable time limits, Office by Office, see www.wipo.int/pct/en/texts/time_limits.html and the *PCT Applicant's Guide*, National Chapters.

   Within **19 months from the priority date, the applicant may request that a supplementary international search be carried out** by a different International Searching Authority that offers this service (Rule 45*bis*.1). The procedure for requesting supplementary international search is described in the *PCT Applicant's Guide*, International Phase, paragraphs 8.006-8.032.

Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300	Authorized officer Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 Telephone No. PCT OSP: 571-272-7774

Form PCT/ISA/220 (July 2014)

Received
JUL 10 2017
Cooley LLP

DOCKETED SMD 7/10/17
DUE ARTICLE 19 AMENDMENTS
DATE 9/6/17
DEADLINE WRITTEN OPINION 2/27/18

PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

To: CHRISTOPHER MAX COLICE
COOLEY LLP
1299 PENNSYLVANIA AVENUE, NW SUITE 700
WASHINGTON, DC 20004-2400

PCT

NOTIFICATION OF TRANSMITTAL OF THE INTERNATIONAL SEARCH REPORT AND THE WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY, OR THE DECLARATION

(PCT Rule 44.1)

	Date of mailing *(day/month/year)* 06 JUL 2017
Applicant's or agent's file reference NRLA00901WO	**FOR FURTHER ACTION** See paragraphs 1 and 4 below
International application No. PCT/US2017/029866	International filing date *(day/month/year)* 27 April 2017
Applicant NEURALA, INC.	

1. ☒ The applicant is hereby notified that the international search report and the written opinion of the International Searching Authority have been established and are transmitted herewith.

   Filing of amendments and statement under Article 19:
   The applicant is entitled, if he so wishes, to amend the claims of the international application (see Rule 46):

   When? The time limit for filing such amendments is normally two months from the date of transmittal of the international search report.

   How? Directly to the International Bureau of WIPO preferably through ePCT or on paper to, 34 chemin des Colombettes 1211 Geneva 20, Switzerland, Facsimile No.: +41 22 338 82 70

   For more detailed instructions, see *PCT Applicant's Guide*, International Phase, paragraphs 9.004 – 9.011.

2. ☐ The applicant is hereby notified that no international search report will be established and that the declaration under Article 17(2)(a) to that effect and the written opinion of the International Searching Authority are transmitted herewith.

3. ☐ **With regard to any protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:
   ☐ the protest together with the decision thereon has been transmitted to the International Bureau together with any request to forward the texts of both the protest and the decision thereon to the designated Offices.
   ☐ no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Reminders**

   The applicant may **submit comments on an informal basis on the written opinion of the International Searching Authority** to the International Bureau. These comments will be made available to the public after international publication. The International Bureau will send a copy of such comments to all designated Offices unless an international preliminary examination report has been or is to be established.

   Shortly after the expiration of **18 months from the priority date, the international application will be published** by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau before the completion of the technical preparations for international publication (Rules 90*bis*.1 and 90*bis*.3).

   Within **19 months** from the priority date, but only in respect of some designated Offices, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase **until 30 months** from the priority date (in some Offices even later); otherwise, the applicant must, **within 20 months** from the priority date, perform the prescribed acts for **entry into the national phase** before those designated Offices. In respect of other designated Offices, the time limit of **30 months** (or later) will apply even if no demand is filed within 19 months. For details about the applicable time limits, Office by Office, see www.wipo.int/pct/en/texts/time_limits.html and the *PCT Applicant's Guide*, National Chapters.

   Within **19 months from the priority date, the applicant may request that a supplementary international search be carried out** by a different International Searching Authority that offers this service (Rule 45*bis*.1). The procedure for requesting supplementary international search is described in the *PCT Applicant's Guide*, International Phase, paragraphs 8.006-8.032.

Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300	Authorized officer Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 Telephone No. PCT OSP: 571-272-7774

Form PCT/ISA/220 (July 2014)

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference NRLA00901WO	FOR FURTHER ACTION	see Form PCT/ISA/220 as well as, where applicable, item 5 below.
International application No. PCT/US2017/029866	International filing date *(day/month/year)* 27 April 2017	(Earliest) Priority Date *(day/month/year)* 27 April 2016
Applicant NEURALA, INC.		

This international search report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This international search report consists of a total of 3 sheets.

☐ It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**

a. With regard to the **language**, the international search was carried out on the basis of:

☒ the international application in the language in which it was filed.

☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

b. ☐ This international search report has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43.6*bis*(a)).

c. ☐ With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, see Box No. I.

2. ☐ **Certain claims were found unsearchable** (see Box No. II).

3. ☐ **Unity of invention is lacking** (see Box No. III).

4. With regard to the **title**,

☒ the text is approved as submitted by the applicant.

☐ the text has been established by this Authority to read as follows:

5. With regard to the **abstract**,

☒ the text is approved as submitted by the applicant.

☐ the text has been established, according to Rule 38.2, by this Authority as it appears in Box No. IV. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. With regard to the **drawings**,

a. the figure of the **drawings** to be published with the abstract is Figure No. 7

☒ as suggested by the applicant.

☐ as selected by this Authority, because the applicant failed to suggest a figure.

☐ as selected by this Authority, because this figure better characterizes the invention.

b. ☐ none of the figures is to be published with the abstract.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/029866

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - B25J 9/16; G05B 15/00; G05B 19/18; G06N 3/00; G06N 3/02; G06N 3/04; G06N 3/08 (2017.01)

CPC - B25J 9/16; B25J 9/163; B25J 9/1697; G05B 15/00; G05B 19/18; G06N 3/00; G06N 3/02; G06N 3/04; G06N 3/049; G06N 3/08 (2017.02)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

USPC - 700/245.000; 700/264.000; 706/15.000; 706/23.000; 706/25.000; 706/26.000; 706/27.000 (keyword delimited)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	US 2016/0096270 A1 (BRAIN CORPORATION) 07 April 2016 (07.04.2016) entire document	1-4, 7-16, 19, 20 --- 5, 6, 17, 18
Y	US 9,177,246 B2 (QUALCOMM TECHNOLOGIES INC.) 03 November 2015 (03.11.2015) entire document	5, 6, 17, 18
A	BERENSON et al. A robot path planning framework that learns from experience. 2012 IEEE International Conference on Robotics and Automation; 1-8 ,2012. [retrieved on 15.06.2017]. Retrieved from the Internet. <URL:http://users.wpi.edu/~dberenson/lightning.pdf>. entire document	1-20
A	US 2016/0075017 A1 (BRAIN CORPORATION) 17 March 2016 (17.03.2016) entire document	1-20
A	US 2015/0127149 A1 (BRAIN CORPORATION) 07 May 2015 (07.05.2015) entire document	1-20
A	US 5,172,253 A (LYNNE) 15 December 1992 (15.12.1992) entire document	1-20
A	US 2014/0032461 A1 (BOARD OF TRUSTEES OF MICHIGAN STATE UNIVERSITY) 30 January 2014 (30.01.2014) entire document	1-20
A	US 2015/0134232 A1 (ROBINSON) 14 May 2015 (14.05.2015) entire document	1-20
A	US 8,392,346 B2 (UEDA et al) 05 March 2013 (05.03.2013) entire document	1-20

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 June 2017	06 JUL 2017
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300	Authorized officer Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/029866

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 9,031,692 B2 (ZHU) 12 May 2015 (12.05.2015) entire document	1-20

PATENT COOPERATION TREATY

From the
INTERNATIONAL SEARCHING AUTHORITY

To: CHRISTOPHER MAX COLICE
COOLEY LLP
1299 PENNSYLVANIA AVENUE, NW SUITE 700
WASHINGTON, DC 20004-2400

PCT

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

(PCT Rule 43*bis*.1)

Date of mailing *(day/month/year)*	06 JUL 2017
Applicant's or agent's file reference NRLA00901WO	**FOR FURTHER ACTION** See paragraph 2 below

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/US2017/029866	27 April 2017	27 April 2016

International Patent Classification (IPC) or both national classification and IPC
IPC(8) - B25J 9/16; G05B 15/00; G05B 19/18; G06N 3/00; G06N 3/02; G06N 3/04; G06N 3/08 (2017.01)
CPC - B25J 9/16; B25J 9/163; B25J 9/1697; G05B 15/00; G05B 19/18; G06N 3/00; G06N 3/02; G06N 3/04; G06N 3/049; G06N 3/08 (2017.02)

Applicant NEURALA, INC.

1. This opinion contains indications relating to the following items:

- [x] Box No. I Basis of the opinion
- [ ] Box No. II Priority
- [ ] Box No. III Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- [ ] Box No. IV Lack of unity of invention
- [x] Box No. V Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step and industrial applicability; citations and explanations supporting such statement
- [ ] Box No. VI Certain documents cited
- [ ] Box No. VII Certain defects in the international application
- [ ] Box No. VIII Certain observations on the international application

2. **FURTHER ACTION**

If a demand for international preliminary examination is made, this opinion will be considered to be a written opinion of the International Preliminary Examining Authority ("IPEA") except that this does not apply where the applicant chooses an Authority other than this one to be the IPEA and the chosen IPEA has notified the International Bureau under Rule 66.1*bis*(b) that written opinions of this International Searching Authority will not be so considered.

If this opinion is, as provided above, considered to be a written opinion of the IPEA, the applicant is invited to submit to the IPEA a written reply together, where appropriate, with amendments, before the expiration of 3 months from the date of mailing of Form PCT/ISA/220 or before the expiration of 22 months from the priority date, whichever expires later.

For further options, see Form PCT/ISA/220.

Name and mailing address of the ISA/US	Date of completion of this opinion	Authorized officer
Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450, Alexandria, VA 22313-1450 Facsimile No. 571-273-8300	20 June 2017	Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/237 (cover sheet) (January 2015)

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US2017/029866

Box No. I Basis of this opinion

1. With regard to the **language**, this opinion has been established on the basis of:

   ☒ the international application in the language in which it was filed.

   ☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

2. ☐ This opinion has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43*bis*.1(a)).

3. ☐ With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, this opinion has been established on the basis of a sequence listing:

   a. ☐ forming part of the international application as filed:

      ☐ in the form of an Annex C/ST.25 text file.

      ☐ on paper or in the form of an image file.

   b. ☐ furnished together with the international application under PCT Rule 13*ter*.1(a) for the purposes of international search only in the form of an Annex C/ST.25 text file.

   c. ☐ furnished subsequent to the international filing date for the purposes of international search only:

      ☐ in the form of an Annex C/ST.25 text file (Rule 13*ter*.1(a)).

      ☐ on paper or in the form of an image file (Rule 13*ter*.1(b) and Administrative Instructions, Section 713).

4. ☐ In addition, in the case that more than one version or copy of a sequence listing has been filed or furnished, the required statements that the information in the subsequent or additional copies is identical to that forming part of the application as filed or does not go beyond the application as filed, as appropriate, were furnished.

5. Additional comments:

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US2017/029866

Box No. V **Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step and industrial applicability; citations and explanations supporting such statement**

1. Statement

Novelty (N)	Claims	5, 6, 17, 18	YES
	Claims	1-4, 7-16, 19, 20	NO
Inventive step (IS)	Claims	None	YES
	Claims	1-20	NO
Industrial applicability (IA)	Claims	1-20	YES
	Claims	None	NO

2. Citations and explanations:

Claims 1-4, 7-16, 19, and 20 lack novelty under PCT Article 33(2) as being anticipated by Brain Corporation (hereinafter Brain).

Regarding claim 1, Brain discloses a computer-implemented method for generating an action for a robot (Para. 126, adaptive predictor 410 and combiner 464 may cooperate to produce a control output 466 useful for operating the robotic device 480 [...] output 466 may comprise one or more motor commands configured to cause one or more actions), the method comprising:
collecting a first experience for the robot (Para. 121, system 440 may comprise an interface component 454 [...] configured to detect instructions; see also Para. 71, rover 322 may comprise an adaptive controller [...] wherein a teacher may train the adaptive controller of the rover 322 [...] the teacher may employ a demonstration [...] using one or more applicable user interfaces), the first experience representing:
a first state of the robot at a first time (Para. 75, upon completion of one or more teacher-guided trials (e.g. first trial) [...] the robot 322 may be configured to perform one or more teacher-assisted trials [...] the user input may comprise reward/penalty signals to the robot. The reward/penalty signal may be based on the robot entering given states (e.g. reward for robot orienting itself towards the target) [interpreted as first state]),
a first action taken by the robot at the first time (Para. 75, reward for robot orienting itself towards the target [interpreted orienting itself as first action]),
a first reward received by the robot in response to the first action (Para. 75, upon completion of one or more teacher-guided trials (e.g. first trial) [...] the robot 322 may be configured to perform one or more teacher-assisted trials [...] the user input may comprise reward/penalty signals to the robot. The reward/penalty signal may be based on the robot entering given states (e.g. reward for robot orienting itself towards the target) [interpreted as first reward]), and
a second state of the robot in response to the first action at a second time after the first time (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal);
determining a degree of similarity between the first experience and a plurality of experiences stored in a memory for the robot (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input);
pruning the plurality of experiences in the memory based on the degree of similarity between the first experience and the plurality of experiences to form a pruned plurality of experiences stored in the memory (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out);
training a neural network associated with the robot with the pruned plurality of experiences (Para. 71, rover 322 may comprise an adaptive controller [...] wherein a teacher may train the adaptive controller of the rover 322 [...] the teacher may employ a demonstration [...] using one or more applicable user interfaces; see also Para. 149, the learning component may be configured to operate a supervised learning process comprising one or more of a neural network); and
generating a second action for the robot using the neural network (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal [interpreted as taking second action]; see also Para. 149, the learning component may be configured to operate a supervised learning process comprising one or more of a neural network).

Regarding claim 2, Brain discloses the computer-implemented method of claim 1, wherein the pruning further comprises:
for each experience in the plurality of experiences (Para. 153, for a given test input):
computing a distance from the first experience (Para. 153, KNN process may be configured to determine closest sample(s) [interpreted as experience] in memory to provide corresponding output (and/or average of outputs). Closeness can be evaluated using Euclidean distance, Manhattan distance); and
comparing the distance to another distance of that experience from each other experience in the plurality of experiences (Para. 153, KNN process may be configured to determine closest sample(s) [interpreted as experience] in memory to provide corresponding output (and/or average of outputs). Closeness can be evaluated using Euclidean distance, Manhattan distance); and
removing a second experience from the memory based on the comparison (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out), the second experience being at least one of the first experience and an experience from the plurality of experiences (Para. 152, the average of all the tree outputs for a sample (or a majority vote if they are discrete classes) may comprise the output of the random forest [interpreted average output as a combination of an experience and at least one of the first experience]).

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.
Continuation of:

Regarding claim 3, Brain discloses the computer-implemented method of claim 2, further comprising removing the second experience from the memory (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out) based on a probability that the distance of the second experience from the first experience and each experience in the plurality of experiences is less than a user-defined threshold (Para. 161, the storing policy may be configured based on the average distance between samples [...] with the smallest distance measure to it [interpreted smallest distance as threshold] [...] the new sample may be stored at random buffer location with probability [...] that may be configured depending on the ratio of the distance of the sample to its nearest neighbors relative to the average distance).

Regarding claim 4, Brain discloses the computer-implemented method of claim 1, where the pruning further includes ranking the first experience and each experience in the plurality of experiences (Para. 152, a given input feature may be compared to a value (e.g., is the feature number 471 greater than 0.7; is feature 223 smaller than 0.3) [interpreted as ranking based on features] [...] the leaves of the tree may correspond to predicted values/classes for the given input; see also Para. 153, the learning component may operate an online KNN learning process comprising a stochastic gradient descent operation [...] configured to store (in a memory buffer) training inputs and outputs).

Regarding claim 7, Brain discloses the computer-implemented method of claim 1, wherein at a first input state the neural network (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal; see also Para. 149, the learning component may be configured to operate a supervised learning process comprising one or more of a neural network) generates an output based at least in part on the pruned plurality of experiences (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out).

Regarding claim 8, Brain discloses the computer-implemented method of claim 1, wherein the pruned plurality of experiences (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out) includes a diverse set of states of the robot (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal).

Regarding claim 9, Brain discloses the computer-implemented method of claim 1, wherein the generating the second action for the robot includes determining that the robot is in the first state and selecting the second action to be different than the first action (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal [interpreted as taking second action]; see also Para. 149, the learning component may be configured to operate a supervised learning process comprising one or more of a neural network).

Regarding claim 10, Brain discloses the computer-implemented method of claim 9, further comprising:
receiving a second reward by the robot in response to the second action (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal [interpreted as taking second action] [...] the reward/penalty signal may be based on the robot entering given states).

Regarding claim 11, Brain discloses the computer-implemented method of claim 1, further comprising:
collecting a second experience for the robot (Para. 121, system 440 may comprise an interface component 454 [...] configured to detect instructions [interpreted as being more than one experience]; see also Para. 71, rover 322 may comprise an adaptive controller [...] wherein a teacher may train the adaptive controller of the rover 322 [...] the teacher may employ a demonstration [...] using one or more applicable user interfaces), the second experience representing:
a second state of the robot (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal),
the second action taken by the robot in response to the second state (Para. 75, the reward/penalty signal may be based on the robot entering given states (e.g. reward for robot orienting itself away from the target) [interpreted as the reward may be given for various actions]),
a second reward received by the robot in response to the second action (Para. 75, the reward/penalty signal may be based on the robot entering given states [interpreted action taken to enter second state may result in another reward]), and
a third state of the robot in response to the second action (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [interpreted other trial as third state] [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal);
determining a degree of similarity between the second experience and the pruned plurality of experiences (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [interpreted the process will be similar for pruning purposes]); and
pruning the pruned plurality of experiences in the memory based on the degree of similarity between the second experience and the pruned plurality of experiences (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out).

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.
Continuation of:

Regarding claim 12, Brain discloses a system for generating a second action for a robot (Fig. 4B, adaptive predictor 410; Para. 126, adaptive predictor 410 and combiner 464 may cooperate to produce a control output 466 useful for operating the robotic device 480 [...] output 466 may comprise one or more motor commands configured to cause one or more actions), the system comprising:
an interface to collect a first experience for the robot (Fig. 4, interface component 454; Para. 121, system 440 may comprise an interface component 454 [...] configured to detect instructions; see also Para. 71, rover 322 may comprise an adaptive controller [...] wherein a teacher may train the adaptive controller of the rover 322 [...] the teacher may employ a demonstration [...] using one or more applicable user interfaces), the first experience representing:
a first state of the robot at a first time (Para. 75, upon completion of one or more teacher-guided trials (e.g. first trial) [...] the robot 322 may be configured to perform one or more teacher-assisted trials [...] the user input may comprise reward/penalty signals to the robot. The reward/penalty signal may be based on the robot entering given states (e.g. reward for robot orienting itself towards the target) [interpreted as first state]),
a first action taken by the robot at the first time (Para. 75, reward for robot orienting itself towards the target [interpreted orienting itself as first action]),
a first reward received by the robot in response to the first action (Para. 75, upon completion of one or more teacher-guided trials (e.g. first trial) [...] the robot 322 may be configured to perform one or more teacher-assisted trials [...] the user input may comprise reward/penalty signals to the robot. The reward/penalty signal may be based on the robot entering given states (e.g. reward for robot orienting itself towards the target) [interpreted as first reward]), and
a second state of the robot in response to the first action at a second time after the first time (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal);
a memory (Fig. 6, memory 614) to store at least one of a plurality of experiences (Para. 77, the learning process may be configured to store intermediate learning stages corresponding to one or more portions of the trajectory traversal) and a pruned plurality of experiences for the robot (Para. 151, the learning component may operate an online random forests learning process [...] samples are accumulated in small batches [...] and a new tree is created for each batch [...] a pruning algorithm is used to discard trees [...] classification happens synchronously);
a processor (Fig. 6, processing components 616; Para. 175, processing component 616 may interface [...] to perform sensory processing), in digital communication with the interface and the memory, to:
determine a degree of similarity between the first experience and the plurality of experiences stored in the memory (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input);
prune the plurality of experiences in the memory based on the degree of similarity between the first experience and the plurality of experiences to form the pruned plurality of experiences (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out);
update the memory to store the pruned plurality of experiences (Para. 151, a pruning algorithm is used to discard trees [...] the learning process may occur asynchronously wherein batches of data may be collected; see also Para. 153, the learning component may operate an online KNN learning process [...] configured to store (in a memory buffer) training inputs and outputs);
train a neural network associated with the robot with the pruned plurality of experiences (Para. 71, rover 322 may comprise an adaptive controller [...] wherein a teacher may train the adaptive controller of the rover 322 [...] the teacher may employ a demonstration [...] using one or more applicable user interfaces; see also Para. 149, the learning component may be configured to operate a supervised learning process comprising one or more of a neural network); and
generate the second action for the robot using the neural network (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal [interpreted as taking second action]; see also Para. 149, the learning component may be configured to operate a supervised learning process comprising one or more of a neural network).

Regarding claim 13, Brain discloses the system of claim 12, further comprising:
a cloud brain (Fig. 4, corrector 424; Para. 86, corrector 424 may comprise a control entity [...] communicating with the robot via a remote controller (wired and/or wireless)) in digital communication with the processor and the robot to transmit the second action to the robot (Para. 174, the learning process may be effectuated by a BrainS software [...] BrainOS [...] may provide functionality [...] that may be typical of an operating system [...] to/from remote cloud storage; see also Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal).

Regarding claim 14, Brain discloses the system of claim 12, wherein the processor is further configured to:
for each experience in the plurality of experiences (Para. 153, for a given test input):
compute a distance from the first experience (Para. 153, KNN process may be configured to determine closest sample(s) [interpreted as experience] in memory to provide corresponding output (and/or average of outputs). Closeness can be evaluated using Euclidean distance, Manhattan distance); and
compare the distance to another distance of that experience from each other experience in the plurality of experiences (Para. 153, KNN process may be configured to determine closest sample(s) [interpreted as experience] in memory to provide corresponding output (and/or average of outputs). Closeness can be evaluated using Euclidean distance, Manhattan distance); and
remove a second experience from the memory based on the comparison (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out), the second experience being at least one of the first experience and an experience from the plurality of experiences (Para. 152, the average of all the tree outputs for a sample (or a majority vote if they are discrete classes) may comprise the output of the random forest [interpreted average output as a combination of an experience and at least one of the first experience]).

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.
Continuation of:

Regarding claim 15, Brain discloses the system of claim 14, wherein the processor is configured to remove the second experience from the memory (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out) based on a probability determination of the distance of the second experience from the first experience and each experience in the plurality of experiences being less than a user-defined threshold (Para. 161, the storing policy may be configured based on the average distance between samples [...] with the smallest distance measure to it [interpreted smallest distance as threshold] [...] the new sample may be stored at random buffer location with probability [...] that may be configured depending on the ratio of the distance of the sample to its nearest neighbors relative to the average distance).

Regarding claim 16, Brain discloses the system of claim 12, wherein the processor is configured to prune the memory based on ranking the first experience and each experience in the plurality of experiences (Para. 152, a given input feature may be compared to a value (e.g., is the feature number 471 greater than 0.7; is feature 223 smaller than 0.3) [...] the leaves of the tree may correspond to predicted values/classes for the given input; see also Para. 153, the learning component may operate an online KNN learning process comprising a stochastic gradient descent operation [...] configured to store (in a memory buffer) training inputs and outputs).

Regarding claim 19, Brain discloses the system of claim 13, wherein at a first input state the neural network (Para. 75, the robot 322 may be configured to perform one or more teacher-assisted trials (e.g. subsequent trials 324, 326, 328) [...] during the trial 324, the robot may progress along the trajectory portion 334 [...] and/or taking certain actions during trajectory traversal; see also Para. 149, the learning component may be configured to operate a supervised learning process comprising one or more of a neural network) generates an output based at least in part on the pruned plurality of experiences (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out).

Regarding claim 20, Brain discloses a computer-implemented method for updating a memory, the memory storing a plurality of experiences (Para. 77, the learning process may be configured to store intermediate learning stages corresponding to one or more portions of the trajectory traversal) received from a computer-based application (Para. 175, computer-readable media containing instructions), the method comprising:
receiving a new experience from the computer-based application (Para. 121, system 440 may comprise an interface component 454 [...] configured to detect instructions; see also Para. 71, rover 322 may comprise an adaptive controller [...] wherein a teacher may train the adaptive controller of the rover 322 [...] the teacher may employ a demonstration [...] using one or more applicable user interfaces);
determining a degree of similarity between the new experience and the plurality of experiences (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input);
adding the new experience based on the degree of similarity (Para. 153, the learning component may operate an online KNN learning process comprising a stochastic gradient descent operation [...] configured to store (in a memory buffer) training inputs and outputs);
removing at least one of the new experience and an experience from the plurality of experiences based on the degree of similarity (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] the trees with smaller weights can be pruned out); and
sending an updated version of the plurality of experiences to the computer-based application (Para. 152, the average of all the tree outputs for a sample (or a majority vote if they are discrete classes) may comprise the output of the random forest [interpreted average output as a combination of an experience and at least one of the first experience]).

Claims 5, 6, 17, and 18 lack an inventive step under PCT Article 33(3) as being obvious over Brain Corporation (hereinafter Brain) in view of QUALCOMM TECHNOLOGIES INC. (hereinafter QUALCOMM).

Regarding claim 5, Brain discloses the computer-implemented method of claim 4, wherein the ranking includes creating a plurality of clusters (Para. 152, a given input feature may be compared to a value (e.g., is the feature number 471 greater than 0.7; is feature 223 smaller than 0.3) [...] the leaves of the tree may correspond to predicted values/classes for the given input [interpreted trees as similar to clusters]) based at least in part on weights and automatically discarding the first experience upon determining that the first experience fits one of the plurality of clusters (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] averaging algorithm could be used where different trees have different weights [...] the trees with smaller weights can be pruned out), but fails to teach weights being synaptic weights. QUALCOMM is in the field of a robotic apparatus (Col. 1 Ln 56-57) and teaches weights being synaptic weights (Col. 9 Ln 22-26, each of the connections [...] is assigned [...] connection efficacy [...] for example a parameter: synaptic weight).

It would have been obvious to one of ordinary skill in the art at the time of the invention wherein weights being synaptic weights as taught by QUALCOMM in the system of Brain. The motivation would have been to provide an improved robotic controller and high in performance (QUALCOMM, Col. 3 Ln 24-26).

Regarding claim 6, modified Brain discloses the computer-implemented method of claim 5, wherein the ranking includes encoding each experience in the plurality of experiences (Para. 152, a given input feature may be compared to a value (e.g., is the feature number 471 greater than 0.7; is feature 223 smaller than 0.3) [...] the leaves of the tree may correspond to predicted values/classes for the given input; see also Para. 188, the output determination may comprise [...] encoding a plurality of features into one or more output classes [interpreted that outputs are encoded]), encoding the first experience (Para. 115, actions of the robot may be encoded into a plurality of discrete states), and comparing the encoded experiences to the plurality of clusters (Para. 152, a given input feature may be compared to a value (e.g., is the feature number 471 greater than 0.7; is feature 223 smaller than 0.3) [...] the leaves of the tree may correspond to predicted values/classes for the given input).

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.
Continuation of:

Regarding claim 17, Brain discloses the system of claim 16, wherein the processor is further configured to:
create a plurality of clusters (Para. 152, a given input feature may be compared to a value (e.g., is the feature number 471 greater than 0.7; is feature 223 smaller than 0.3) [...] the leaves of the tree may correspond to predicted values/classes for the given input [interpreted trees as similar to clusters]) based at least in part on weights (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] averaging algorithm could be used where different trees have different weights [...] the trees with smaller weights can be pruned out);
rank the first experience and the plurality of experiences based on the plurality of clusters (Para. 152, a given input feature may be compared to a value (e.g., is the feature number 471 greater than 0.7; is feature 223 smaller than 0.3) [interpreted as ranking based on features] [...] the leaves of the tree may correspond to predicted values/classes for the given input [interpreted trees as similar to clusters]); and
automatically discard the first experience upon determination that the first experience fits one of the plurality of clusters (Para. 152, a given input feature may be compared to a value [...] the leaves of the tree may correspond to predicted values/classes for the given input [...] oldest tress may be discarded [...] averaging algorithm could be used where different trees have different weights [...] the trees with smaller weights can be pruned out), but fails to teach weights being synaptic weights. QUALCOMM is in the field of a robotic apparatus (Col. 1 Ln 56-57) and teaches weights being synaptic weights (Col. 9 Ln 22-26, each of the connections [...] is assigned [...] connection efficacy [...] for example a parameter: synaptic weight).

It would have been obvious to one of ordinary skill in the art at the time of the invention wherein weights being synaptic weights as taught by QUALCOMM in the system of Brain. The motivation would have been to provide an improved robotic controller and high in performance (QUALCOMM, Col. 3 Ln 24-26).

Regarding claim 18, modified Brain discloses the system of claim 17, wherein the processor is further configured to encode each experience in the plurality of experiences (Para. 152, a given input feature may be compared to a value (e.g., is the feature number 471 greater than 0.7; is feature 223 smaller than 0.3) [...] the leaves of the tree may correspond to predicted values/classes for the given input; see also Para. 188, the output determination may comprise [...] encoding a plurality of features into one or more output classes [interpreted that outputs are encoded]), encode the first experience (Para. 115, actions of the robot may be encoded into a plurality of discrete states), and compare the encoded experiences to the plurality of clusters (Para. 152, a given input feature may be compared to a value (e.g., is the feature number 471 greater than 0.7; is feature 223 smaller than 0.3) [...] the leaves of the tree may correspond to predicted values/classes for the given input).

Claims 1-20 meet the criteria set out in PCT Article 33(4), and thus have industrial applicability because the subject matter claimed can be made or used in industry.

PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

To: JOSEPH TEJA
COOLEY LLP
1299 PENNSYLVANIA AVE.
SUITE 700
WASHINGTON, DC 20004

PCT

NOTIFICATION OF TRANSMITTAL OF THE INTERNATIONAL SEARCH REPORT AND THE WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY, OR THE DECLARATION

(PCT Rule 44.1)

	Date of mailing *(day/month/year)*
Applicant's or agent's file reference NRLA00301WO	**FOR FURTHER ACTION** See paragraphs 1 and 4 below
International application No. PCT/US 14/39239	International filing date *(day/month/year)* 22 May 2014 (22.05.2014)
Applicant NEURALA, INC.	

1. ☒ The applicant is hereby notified that the international search report and the written opinion of the International Searching Authority have been established and are transmitted herewith.

   Filing of amendments and statement under Article 19:
   The applicant is entitled, if he so wishes, to amend the claims of the international application (see Rule 46):

   When? The time limit for filing such amendments is normally two months from the date of transmittal of the international search report.

   How? Directly to the International Bureau of WIPO preferably through ePCT or on paper to, 34 chemin des Colombettes 1211 Geneva 20, Switzerland, Facsimile No.: +41 22 338 82 70

   For more detailed instructions, see *PCT Applicant's Guide*, International Phase, paragraphs 9.004 – 9.011.

2. ☐ The applicant is hereby notified that no international search report will be established and that the declaration under Article 17(2)(a) to that effect and the written opinion of the International Searching Authority are transmitted herewith.

3. ☐ **With regard to any protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:
   ☐ the protest together with the decision thereon has been transmitted to the International Bureau together with any request to forward the texts of both the protest and the decision thereon to the designated Offices.
   ☐ no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Reminders**

   The applicant may **submit comments on an informal basis on the written opinion of the International Searching Authority** to the International Bureau. These comments will be made available to the public after international publication. The International Bureau will send a copy of such comments to all designated Offices unless an international preliminary examination report has been or is to be established.

   Shortly after the expiration of **18 months from the priority date, the international application will be published** by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau before the completion of the technical preparations for international publication (Rules 90*bis*.1 and 90*bis*.3).

   Within **19 months** from the priority date, but only in respect of some designated Offices, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase **until 30 months** from the priority date (in some Offices even later); otherwise, the applicant must, **within 20 months** from the priority date, perform the prescribed acts for **entry into the national phase** before those designated Offices. In respect of other designated Offices, the time limit of **30 months** (or later) will apply even if no demand is filed within 19 months. For details about the applicable time limits, Office by Office, see www.wipo.int/pct/en/texts/time_limits.html and the *PCT Applicant's Guide*, National Chapters.

   Within **19 months from the priority date, the applicant may request that a supplementary international search be carried out** by a different International Searching Authority that offers this service (Rule 45*bis*.1). The procedure for requesting supplementary international search is described in the *PCT Applicant's Guide*, International Phase, paragraphs 8.006-8.032.

Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer Lee W. Young PCT Helpdesk: 571-272-4300 Telephone No. PCT OSP: 571-272-7774

Form PCT/ISA/220 (July 2014)

R.19 Omnd 1-26-2015
R.34 Omnd 3-22-2015

PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

To: JOSEPH TEJA
COOLEY LLP
1299 PENNSYLVANIA AVE.
SUITE 700
WASHINGTON, DC 20004

PCT

NOTIFICATION OF TRANSMITTAL OF THE INTERNATIONAL SEARCH REPORT AND THE WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY, OR THE DECLARATION

(PCT Rule 44.1)

	Date of mailing *(day/month/year)* 26 NOV 2014
Applicant's or agent's file reference NRLA00301WO	**FOR FURTHER ACTION** See paragraphs 1 and 4 below
International application No. PCT/US 14/39239	International filing date *(day/month/year)* 22 May 2014 (22.05.2014)
Applicant NEURALA, INC.	

1. ☒ The applicant is hereby notified that the international search report and the written opinion of the International Searching Authority have been established and are transmitted herewith.

   Filing of amendments and statement under Article 19:
   The applicant is entitled, if he so wishes, to amend the claims of the international application (see Rule 46):

   When? The time limit for filing such amendments is normally two months from the date of transmittal of the international search report.

   How? Directly to the International Bureau of WIPO preferably through ePCT or on paper to, 34 chemin des Colombettes 1211 Geneva 20, Switzerland, Facsimile No.: +41 22 338 82 70

   For more detailed instructions, see *PCT Applicant's Guide*, International Phase, paragraphs 9.004 – 9.011.

2. ☐ The applicant is hereby notified that no international search report will be established and that the declaration under Article 17(2)(a) to that effect and the written opinion of the International Searching Authority are transmitted herewith.

3. ☐ **With regard to any protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:
   - ☐ the protest together with the decision thereon has been transmitted to the International Bureau together with any request to forward the texts of both the protest and the decision thereon to the designated Offices.
   - ☐ no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Reminders**

   The applicant may **submit comments on an informal basis on the written opinion of the International Searching Authority** to the International Bureau. These comments will be made available to the public after international publication. The International Bureau will send a copy of such comments to all designated Offices unless an international preliminary examination report has been or is to be established.

   Shortly after the expiration **of 18 months from the priority date, the international application will be published** by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau before the completion of the technical preparations for international publication (Rules 90*bis*.1 and 90*bis*.3).

   Within **19 months** from the priority date, but only in respect of some designated Offices, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase **until 30 months** from the priority date (in some Offices even later); otherwise, the applicant must, **within 20 months** from the priority date, perform the prescribed acts for **entry into the national phase** before those designated Offices. In respect of other designated Offices, the time limit of **30 months** (or later) will apply even if no demand is filed within 19 months. For details about the applicable time limits, Office by Office, see www.wipo.int/pct/en/texts/time_limits.html and the *PCT Applicant's Guide*, National Chapters.

   Within **19 months from the priority date, the applicant may request that a supplementary international search be carried out** by a different International Searching Authority that offers this service (Rule 45*bis*.1). The procedure for requesting supplementary international search is described in the *PCT Applicant's Guide*, International Phase, paragraphs 8.006-8.032.

Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer Lee W. Young PCT Helpdesk: 571-272-4300 Telephone No. PCT OSP: 571-272-7774

Form PCT/ISA/220 (July 2014)

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference NRLA0030IWO	FOR FURTHER ACTION	see Form PCT/ISA/220 as well as, where applicable, item 5 below.
International application No. PCT/US 14/39239	International filing date *(day/month/year)* 22 May 2014 (22.05.2014)	(Earliest) Priority Date *(day/month/year)* 22 May 2013 (22.05.2013)
Applicant NEURALA, INC.		

This international search report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This international search report consists of a total of 5 sheets.

☐ It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**

a. With regard to the **language**, the international search was carried out on the basis of:

☒ the international application in the language in which it was filed.

☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

b. ☐ This international search report has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43.6*bis*(a)).

c. ☐ With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, see Box No. I.

2. ☐ **Certain claims were found unsearchable** (see Box No. II).

3. ☒ **Unity of invention is lacking** (see Box No. III).

4. With regard to the **title**,

☒ the text is approved as submitted by the applicant.

☐ the text has been established by this Authority to read as follows:

5. With regard to the **abstract**,

☒ the text is approved as submitted by the applicant.

☐ the text has been established, according to Rule 38.2, by this Authority as it appears in Box No. IV. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. With regard to the **drawings**,

a. the figure of the **drawings** to be published with the abstract is Figure No. 2B

☒ as suggested by the applicant.

☐ as selected by this Authority, because the applicant failed to suggest a figure.

☐ as selected by this Authority, because this figure better characterizes the invention.

b. ☐ none of the figures is to be published with the abstract.

INTERNATIONAL SEARCH REPORT

International application No
PCT/US 14/39239

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fees must be paid.

Group I: Claims 1-13, directed to a method for aggregating data representing a moving body with respect to an environment.

Group II: Claims 14-23, directed to a system for navigating a robot through an environment.

Group III: Claims 24-31, directed to a method of providing an estimate of at least one of a position of a moving body and an orientation of the moving body.

--- (See Continuation in Supplemental Box) ---

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:
1-13

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☐ No protest accompanied the payment of additional search fees.

NVIDIA Ex1002 Page 101 of 1081

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 14/39239

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G01C 17/00 (2014.01)
CPC - G06F 3/011
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC (8) - G01C 17/00 (2014.01)
CPC - G06F 3/011

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
CPC - G01C 15/00; G01C 21/20 (See keywords below)
USPC - 702/150, 702/188, 709/201, 340/286.02, 700/82, 340/607, 700/3, 340/606, 701/23, 901/47

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Thomsoninnovation.com; Patbase; Google Scholar; Google Patents; Gogole.com; Freepatentsonline; ProQuest Dialog
Search Terms: Moving object, body, motion, displacement, movement, inertia, acceleration, sensor, data, measurement, reading, aggregate, sum, average, weight, cell, node, map, associate, neural, second, tier, level, graph, resolution, eca

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2007/0198222 A1 (SCHUSTER et al.), 23 August 2007 (23.08.2007), entire document, especially Abstract; para [0025], [0030], [0040]-[0041], [0057]-[0058]	1-13
Y	US 2008/0256880 A1 (SMITH et al.), 23 October 2008 (23.10.2008), entire document, especially Abstract; para [0009], [0052], [0074], [0083], [0095]	1-13
Y	US 6,336,051 B1 (PANGELS et al.), 01 January 2002 (01.01.2002), entire document, especially Abstract; col 6, ln 1-30; col 10, ln 10-40; col 18, ln 15-65; col 19, ln 10-60	7-11
Y	US 2008/0033897 A1 (LLOYD), 07 February 2008 (07.02.2008), entire document, especially Abstract; para [0029]-[0031], [0076]-[0079], [0113]-[0116], [0174]-[0177], [0192]-[0195], [0200]-[0203]	12
A	US 2011/0173015 A1 (Chapman), 14 July 2011 (14.07.2011), entire document	1-13
A	US 2003/0078754 A1 (HAMZA), 24 April 2003 (24.04.2003), entire document	1-13

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 November 2014 (13.11.2014)	26 NOV 2014
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (July 2009)

Continuation of:
Box III. Observations where unity of invention is lacking

Group III: Claims 32-33, directed to a method for developing a robust noise-tolerant distributed data representation model of an environment based on data from at least one sensor.

The inventions listed as Groups I-IV do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons:

Special Technical Features

The invention of Group I included the features of the plurality of sets of data readings resenting independent measurements by the at least one sensor of the moving body with respect to the environment; mapping the plurality of sets of data readings received to respective first sets of data cells, each first set of data cells forming a corresponding first neural representation of the moving body with respect to the environment; mapping the plural of aggregated data readings found to a second set data cells stored in the memory, the second set of cells forming a second neural representation of the moving body with respect to the environment that is more precise than the first neural representations of the moving body with respect to the environment, not required in any other groups.

The invention of Group II included the features of an interface to receive first sensor data from a first sensor and second sensor data from a second sensor, the first sensor data and the second sensor data representing measurements of the environment, with respect to the robot; a memory operably coupled to the interface, to store the first sensor data in a first neural representation of possible readings from the first sensor and to store the second sensor data in a second neural representation of possible readings from the second sensor; and a path integration module operably coupled to the memory, to; apply a first weight to the first sensor data in the first neural representation, the first weight representing a first measurement precision of the first sensor; apply a second weight to the second sensor data in the second neural representation, the Second weight representing a first measurement precision of the first sensor, not required in any other groups.

The invention of Group III included the features of receiving a vestibular signal representative of the at least one of the orientation of the moving body and the position of the moving body; receiving a motor outflow signal representative of commanded motion of the moving body; receiving an optical flow signal representative of the at least of the orientation of the moving body and the position of the moving body with respect to an environment of the moving body, not required in any other groups.

The invention of Group IV included the features of generating, with a processor at least one set of data cells representing respective data points in a measurement range of the at least one sensor, the respective data points representing a continuous portion of the environment; defining, with the processor, a resolution of the neural re-presentation responding to the at least one set of data cells based at least in part on a range of is corresponding set of data readings and on a number of data cells in the at least one set of data cells; determining, with the processor, a scale of the neural representation corresponding to the at least one set of data cells based at least in part on the range of the corresponding set of data readings; and generating with the processor, a set of graph connections among the at least one set of data cells, the set of graph connections providing continuity between representations of the respective data points, not required in any other groups.

Common Technical Features

Groups I-II and IV share the technical features of at least one sensor, sensor data readings, neural representation of the reading.

Groups I-III share the technical feature of a weighting the signal data, combining the values weighted to form a plurality of aggregated data.

Groups I and III share the technical feature of a moving body and plurality of sets of data readings of the moving body or robot with respect to the environment.

Groups I-II share the technical feature of assigning a weight to the sensor data, the first weight representing a first measurement precision of the first sensor.

Groups II-III share the technical feature of estimating a position of the robot or moving body.

Groups I and IV share the technical feature of data cells.

However, the shared technical features do not represent a contribution over prior art as being anticipated by US 2011/0173015 A1 (Chapman), 14 July 2011 (14.07.2011). Chapman teaches at least one sensor, sensor data readings, neural representation of the reading (para [0032], [0077], [0103] - data samples provided by mobile data sources (e.g., vehicles), data readings from road-based traffic sensors, various information about a current data reading distribution is provided as input to a sensor health (or data reading reliability) classifier, such as based on a neural network).

----------(See next page)----------

NVIDIA Ex1002 Page 103 of 1081

Chapman further teaches the groups I-III shared technical feature of a weighting the signal data, combining the values weighted to form a plurality of aggregated data (para [0048], [0052], [0054], [0062] - aggregating data samples , weighted by one or more attributes of the data samples (e.g., age, such as to give greater weight to newer data samples, and/or source or type of the data samples, such as to vary the weight for data samples from mobile data sources or from road sensors so as to give greater weight to sources with higher expected reliability or availability) or by other factors, weighted analysis to reflect a degree of match for each type of information for a data sample to a candidate road segment, Sensor Data Aggregator component 110 aggregates sensor-based traffic condition information provided by the Sensor Data Conditioner component 105), groups I and III shared technical feature of a moving body and plurality of sets of data readings of the moving body or robot with respect to the environment (abstract- data samples from vehicles and other mobile data sources traveling on the roads), groups I-II shared technical feature of assigning a weight to the sensor data, the first weight representing a first measurement precision of the first sensor (para [0048], [0052], [0054], [0062] - weighted by one or more attributes of the data samples (e.g., age, such as to give greater weight to newer data samples, and/or source or type of the data samples, such as to vary the weight for data samples from mobile data sources or from road sensors so as to give greater weight to sources with higher expected reliability or availability) or by other factors, weighted analysis to reflect a degree of match for each type of information for a data sample to a candidate road segment), groups II-III shared technical feature of estimating a position of the robot or moving body (para [0036] - determining the geographic location, speed, direction, and/or other data that characterizes or is otherwise related to the vehicle's travel) and groups I and IV shared technical feature of data cells (para [0162], [0170]- data points).

As the common features were known in the art at the time of the invention, this cannot be considered a common technical feature that would otherwise unify the groups.

Therefore, Groups I-IV lack unity under PCT Rule 13.

NVIDIA Ex1002 Page 104 of 1081

PATENT COOPERATION TREATY

From the
INTERNATIONAL SEARCHING AUTHORITY

To: JOSEPH TEJA
COOLEY LLP
1299 PENNSYLVANIA AVE.
SUITE 700
WASHINGTON, DC 20004

PCT

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

(PCT Rule 43*bis*.1)

Date of mailing *(day/month/year)* 26 NOV 2014

Applicant's or agent's file reference
NRLA00301WO

FOR FURTHER ACTION
See paragraph 2 below

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/US 14/39239	22 May 2014 (22.05.2014)	22 May 2013 (22.05.2013)

International Patent Classification (IPC) or both national classification and IPC
IPC(8) - G01C 17/00 (2014.01)
CPC - G06F 3/011

Applicant NEURALA, INC.

1. This opinion contains indications relating to the following items:

- [X] Box No. I Basis of the opinion
- [ ] Box No. II Priority
- [ ] Box No. III Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- [X] Box No. IV Lack of unity of invention
- [X] Box No. V Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- [ ] Box No. VI Certain documents cited
- [ ] Box No. VII Certain defects in the international application
- [ ] Box No. VIII Certain observations on the international application

2. FURTHER ACTION

If a demand for international preliminary examination is made, this opinion will be considered to be a written opinion of the International Preliminary Examining Authority ("IPEA") except that this does not apply where the applicant chooses an Authority other than this one to be the IPEA and the chosen IPEA has notified the International Bureau under Rule 66.1 *bis*(b) that written opinions of this International Searching Authority will not be so considered.

If this opinion is, as provided above, considered to be a written opinion of the IPEA, the applicant is invited to submit to the IPEA a written reply together, where appropriate, with amendments, before the expiration of 3 months from the date of mailing of Form PCT/ISA/220 or before the expiration of 22 months from the priority date, whichever expires later.

For further options, see Form PCT/ISA/220.

Name and mailing address of the ISA/US	Date of completion of this opinion	Authorized officer:
Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	13 November 2014 (13.11.2014)	Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/237 (cover sheet) (July 2011)

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 14/38239

Box No. I Basis of this opinion

1. With regard to the **language**, this opinion has been established on the basis of:

   ☒ the international application in the language in which it was filed.

   ☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

2. ☐ This opinion has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43*bis*.1(a))

3. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, this opinion has been established on the basis of a sequence listing filed or furnished:

   a. (means)

      ☐ on paper

      ☐ in electronic form

   b. (time)

      ☐ in the international application as filed

      ☐ together with the international application in electronic form

      ☐ subsequently to this Authority for the purposes of search

4. ☐ In addition, in the case that more than one version or copy of a sequence listing has been filed or furnished, the required statements that the information in the subsequent or additional copies is identical to that in the application as filed or does not go beyond the application as filed, as appropriate, were furnished.

5. Additional comments:

NVIDIA Ex1002 Page 106 of 1081

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 14/39239

Box No. IV Lack of unity of invention

1. ☒ In response to the invitation (Form PCT/ISA/206) to pay additional fees the applicant has, within the applicable time limit:

   ☐ paid additional fees.

   ☐ paid additional fees under protest and, where applicable, the protest fee.

   ☐ paid additional fees under protest but the applicable protest fee was not paid.

   ☒ not paid additional fees.

2. ☐ This Authority found that the requirement of unity of invention is not complied with and chose not to invite the applicant to pay additional fees.

3. This Authority considers that the requirement of unity of invention in accordance with Rule 13.1, 13.2 and 13.3 is

   ☐ complied with.

   ☒ not complied with for the following reasons:

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fees must be paid.

Group I: Claims 1-13, directed to a method for aggregating data representing a moving body with respect to an environment.

Group II: Claims 14-23, directed to a system for navigating a robot through an environment.

Group III: Claims 24-31, directed to a method of providing an estimate of at least one of a position of a moving body and an orientation of the moving body.

Group III: Claims 32-33, directed to a method for developing a robust noise-tolerant distributed data representation model of an environment based on data from at least one sensor.

The inventions listed as Groups I-IV do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons:

Special Technical Features

The invention of Group I included the features of the plurality of sets of data readings resenting independent measurements by the at least one sensor of the moving body with respect to the environment; mapping the plurality of sets of data readings received to respective first sets of data cells, each first set of data cells forming a corresponding first neural representation of the moving body with respect to the environment; mapping the plural of aggregated data readings found to a second set data cells stored in the memory, the second set of cells forming a second neural representation of the moving body with respect to the environment that is more precise than the first neural representations of the moving body with respect to the environment, not required in any other groups.

The invention of Group II included the features of an interface to receive first sensor data from a first sensor and second sensor data from a second sensor, the first sensor data and the second sensor data representing measurements of the environment, with respect to the robot; a memory operably coupled to the interface, to store the first sensor data in a first neural representation of possible readings from the first sensor and to store the second sensor data in a second neural representation of possible readings from the second sensor; and a path integration module operably coupled to the memory, to, apply a first weight to the first sensor data in the first neural representation, the first weight representing a first measurement precision of the first sensor; apply a second weight to the second sensor data in the second neural representation, the Second weight representing a first measurement precision of the first sensor, not required in any other groups.

The invention of Group III included the features of receiving a vestibular signal representative of the at least one of the orientation of the moving body and the position of the moving body; receiving a motor outflow signal representative of commanded motion of the moving body; receiving an optical flow signal representative of the at least of the orientation of the moving body and the position of the moving body with respect to an environment of the moving body, not required in any other groups.

--- (See Continuation in Supplemental Box) ---

4. Consequently, this opinion has been established in respect of the following parts of the international application:

   ☐ all parts.

   ☒ the parts relating to claims Nos. 1-13

NVIDIA Ex1002 Page 107 of 1081

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 14/35239

Box No. V **Reasoned statement under Rule 43bis.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement**

1. Statement

Novelty (N)	Claims	1-13	YES
	Claims	None	NO
Inventive step (IS)	Claims	None	YES
	Claims	1-13	NO
Industrial applicability (IA)	Claims	1-13	YES
	Claims	None	NO

2. Citations and explanations:

Claims 1-6 and 13 lack an inventive step under PCT Article 33(3) as being obvious over US 2007/0198222 A1 to Schuster et al. (hereinafter 'Schuster') in view of US 2008/0256880 A1 to Smith et al. (hereinafter 'Smith').

Regarding claim 1, Schuster teaches a method for aggregating data representing an object with respect to an environment (Abstract), the method comprising: receiving, by a processor, a plurality of sets of data readings from at least one sensor, the plurality of sets of readings representing independent measurements by the at least one sensor of the object with respect to the environment (Abstract; para [0025], [0040]-[0041]- plurality of sensors can provide redundant detected data, data from multiple sensors within an industrial automation environment. The system 100 can include any number of sensors 102-104 (e.g., sensor 1 102, . . . sensor N 104, where N can be any positive integer); mapping the plurality of sets of data readings received to respective first sets of data cells, each first set of data cells forming a corresponding first neural representation of the object with respect to the environment (para [0050]-[0053]- an input attribute vector, x=(x1, x2, x3, x4, xn), to a confidence that the input belongs to a class, that is, f(x)=confidence(class). Such classification can employ a probabilistic and/or statistical-based analysis (e.g., factoring into the analysis utilities and costs) to prognose or infer an action that a user desires to be automatically performed. A support vector machine (SVM) is an example of a classifier that can be employed. model classification approaches include, e.g., naive Bayes, Bayesian networks, decision trees, neural networks); applying respective weights to values in the respective first sets of data, the respective weights representing respective precisions of the independent measurements represented by the corresponding sets of data readings (para [0028], [0038]-[0040]- the weighting component 106 can weight the detected data based upon a likelihood of accurate measurement related to each of the sensors 102-104. For instance, each of the sensors 102-104 can provide detected data to the weighting component 106. Additionally, a number of factors related to the sensors 102-104 can be evaluated to generate corresponding weights.); combining the values in the first sets of data weighted to form an aggregated data reading (para [0030], [0057]-[0058] - combination component 105 can be coupled to the weighting component 106 and can generate an output of harmonized data by aggregating the detected data and corresponding assigned weights and/or weighted data provided by the weighting component 106, generate aggregated data from multiple, redundant sensors); processing the data to form a more precise data representations of the moving body with respect to the environment (para [0028]- weight yielded with the weighting component 106 related to the sensor 1 102 can be associated with a confidence that the sensor 1 102 can provided an accurate evaluation of a signal). Schuster does not teach the data is a data representing a moving body with respect to an environment; mapping the plural of aggregated data readings formed in to a second set data cells stored in the memory, the second set of cells forming a second neural representation of the moving body with respect to the environment that is more precise than the first neural representations of the moving body with respect to the environment. However, Smith teaches data representing a moving body with respect to an environment (para [0009], [0052], [0083], [0095] - network system for collecting information from sensors, vehicle-based nodes, movement of a vehicle of interest including information specific to the vehicle. The node 23 autonomously collects external data on the vehicle of interest and the person of interest for matching); associating the plural of aggregated data readings formed in to a second set data nodes in the network, the second set of nodes forming a second neural representation of the moving body with respect to the environment (para [0035], [0052], [0074]- tiered structure provides a subordinate-superordinate relationship of different tiers where the fourth tier network is the most subordinate network and the first tier network is the most superordinate network, collect data from other network nodes within the tier network and to direct the collected data to a center node of a superordinate tier network, topmost node 18 is connected to a set of nodes in the level below it, in a one-to-many relationship. for one national node 18, there is a set of regional nodes 20, The node 23, for example, has its database for storing data, a sensor fusion module for aggregate received sensor data from different sensor nodes 26, and a first-order reasoning module to analyze received data, any intelligent programming paradigm may be incorporated alone, in combination or hybridized, such as genetic algorithms, neural networks). It would have been obvious to one of ordinary skill in the art to combine the teachings of Schuster with those of Smith since doing so would enhance the accuracy of sensor data of a moving object with respect to the environment.

Regarding claim 2, Smith teaches receiving data readings representative of an orientation, motion, and/or position of the moving body with respect to the environment (para [0063]-[0065], [0083] . motion, position).

Regarding claim 3, Schuster teaches processing data from a single sensor to form the first plurality of data readings and the second plurality of data readings by, extracting different modalities of data reading from the data from the single sensor (para [0026]-[0026], [0039] -[0040]).

--- (See Continuation in Supplemental Box) ---

NVIDIA Ex1002 Page 108 of 1081

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.

Continuation of:
Box No. IV Lack of unity of invention

The invention of Group IV included the features of generating, with a processor at least one set of data cells representing respective data points in a measurement range of the at least one sensor, the respective data points representing a continuous portion of the environment; defining, with the processor, a resolution of the neural re-presentation responding to the at least one set of data cells based at least in part on a range of is corresponding set of data readings and on a number of data cells in the at least one set of data cells; determining, with the processor, a scale of the neural representation corresponding to the at least one set of data cells based at least in part on the range of the corresponding set of data readings; and generating with the processor, a set of graph connections among the at least one set of data cells, the set of graph connections providing continuity between representations of the respective data points, not required in any other groups.

Common Technical Features

Groups I-II and IV share the technical features of at least one sensor, sensor data readings, neural representation of the reading.

Groups I-III share the technical feature of a weighting the signal data, combining the values weighted to form a plurality of aggregated data.

Groups I and III share the technical feature of a moving body and plurality of sets of data readings of the moving body or robot with respect to the environment.

Groups I-II share the technical feature of assigning a weight to the sensor data, the first weight representing a first measurement precision of the first sensor.

Groups II-III share the technical feature of estimating a position of the robot or moving body.

Groups I and IV share the technical feature of data cells.

However, the shared technical features do not represent a contribution over prior art as being anticipated by US 2011/0173015 A1 (Chapman), 14 July 2011 (14.07.2011). Chapman teaches at least one sensor, sensor data readings, neural representation of the reading (para [0032], [0077], [0103] - data samples provided by mobile data sources (e.g., vehicles), data readings from road-based traffic sensors, various information about a current data reading distribution is provided as input to a sensor health (or data reading reliability) classifier, such as based on a neural network).

Chapman further teaches the groups I-III shared technical feature of a weighting the signal data, combining the values weighted to form a plurality of aggregated data (para [0048], [0052], [0054], [0062] - aggregating data samples , weighted by one or more attributes of the data samples (e.g , age, such as to give greater weight to newer data samples, and/or source or type of the data samples, such as to vary the weight for data samples from mobile data sources or from road sensors so as to give greater weight to sources with higher expected reliability or availability) or by other factors, weighted analysis to reflect a degree of match for each type of information for a data sample to a candidate road segment, Sensor Data Aggregator component 110 aggregates sensor-based traffic condition information provided by the Sensor Data Conditioner component 105), groups I and III shared technical feature of a moving body and plurality of sets of data readings of the moving body or robot with respect to the environment (abstract- data samples from vehicles and other mobile data sources traveling on the roads), groups I-II shared technical feature of assigning a weight to the sensor data, the first weight representing a first measurement precision of the first sensor (para [0048], [0052], [0054], [0062] - weighted by one or more attributes of the data samples (e.g., age, such as to give greater weight to newer data samples, and/or source or type of the data samples, such as to vary the weight for data samples from mobile data sources or from road sensors so as to give greater weight to sources with higher expected reliability or availability) or by other factors, weighted analysis to reflect a degree of match for each type of information for a data sample to a candidate road segment), groups II-III shared technical feature of estimating a position of the robot or moving body (para [0036] - determining the geographic location, speed, direction, and/or other data that characterizes or is otherwise related to the vehicle's travel) and groups I and IV shared technical feature of data cells (para [0162], [0170]- data points).

As the common features were known in the art at the time of the invention, this cannot be considered a common technical feature that would otherwise unify the groups.

Therefore, Groups I-IV lack unity under PCT Rule 13.

NVIDIA Ex1002 Page 109 of 1081

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.

Continuation of
Box V.2 Citations and Explanations.

Regarding claim 4, Schuster teaches receiving the first plurality of data readings from a first sensor and the second plurality of data readings from a second sensor (para [0039]-[0041], [0048]-[0050]).

Regarding claim 5, Schuster teaches receiving the first plurality of data readings from at least one of a motor, a motor controller, an accelerometer, a gyroscope, an inertial measurement unit, an imaging sensor, a proximity sensor, a photosensor, a sonar, a lidar, a radar, a rangefinder, and a microphone (para [0026]-[0028]- proximity sensor).

Regarding claim 6, Schuster teaches the first plurality of data readings represents at least one of a motor outflow, an optic flow, an inertial measurement Unit measurement, a color measurement, a proximity measurement, a luminance measurement, a velocity measurement, an orientation measurement, an acceleration measurement, and a distance measurement (para [0026]-[0028]- proximity measurement).

Regarding claim 13, Schuster teaches adjusting data readings received from the at least one sensor based at least in part on the plurality of aggregated data readings (para [0028]-[0030]).

Claims 7-11 lack an inventive step under PCT Article 33(3) as being obvious over Schuster in view of Smith and further in view of US 6,336,051 B1 to Pangels et al. (hereinafter 'Pangels').

Regarding claim 7, neither Schuster nor Smith teaches receiving a first set of data readings representing measurements by an inertial measurement unit of the moving body; receiving a second set of data readings representing commands issued to a motor of the moving body; and receiving a third set of data readings representing an optic flow detected by an imaging sensor of the moving body. However, Pangels teaches receiving a first set of data readings representing measurements by an inertial measurement unit of the moving body (Abstract; col 4, ln 55-65); receiving a second set of data readings representing commands issued to a motor of the moving body (col 6, ln 1-30; col 10, ln 10-40); and receiving a third set of data readings representing an optic flow detected by an imaging sensor of the moving body (col 8, ln 5-30; col 10, ln 35-50). It would have been obvious to one of ordinary skill in the art to combine the teachings of Schuster with those of Smith and Pangels since doing so would collect and process various sensor data for controlling the device.

Regarding claim 8, Pangels teaches applying a first weight to the first set of data readings, a second weight to the second set of data readings, and a third weight to the third set of data readings (col 4, ln 55-65; col 18, ln 15-65; col 19, ln 10-60); increasing the first weight relative to the second weight and the third weight for data reading indicative of an increasing speed of the moving body (col 6, ln 1-30; col 10, ln 10-40; col 18, ln 15-65; col 19, ln 10-60); increasing the third weight relative to the first weight and the second weight for data reading indicative of a decreasing speed of the moving body (col 8, ln 5-30; col 10, ln 35-50; col 18, ln 15-65; col 19, ln 10-60), and varying the second weight from a value of zero for data readings indicative of no motion of the moving body to a value between the first weight and the third weight for data readings indicative of motion of the moving body (col 18, ln 15-65, col 19, ln 10-60).

Regarding claim 9, Smith teaches estimating the position and the orientation of the moving body with respect to the environment based on the second neural representation (para [0009], [0035], [0052], [0083], [0095]).

Regarding claim 10, neither Schuster nor Smith teaches sensor data from an imaging sensor associated with the moving body; deriving a first set of data readings from the sensor data, the first set of data readings representing at least one edge sensed by the imaging sensor; deriving a second set of data readings from the sensor data, the second set of data readings representing an estimate of distance between the moving body and at least one object in the environment; and deriving a third set of data readings from the sensor data, the third set of data readings representing motion of the moving body. However, Pangels teaches sensor data from an imaging sensor associated with the moving body (Abstract; col 4, ln 55-65); deriving a first set of data readings from the sensor data, the first set of data readings representing at least one edge sensed by the imaging sensor (col 3, ln 15-60; col 14, ln 1-30); deriving a second set of data readings from the sensor data, the second set of data readings representing an estimate of distance between the moving body and at least one object in the environment (col 8, ln 45-65; col 14, ln 1-35); and deriving a third set of data readings from the sensor data, the third set of data readings representing motion of the moving body (col 8, ln 15-50; col 8, ln 1-45). It would have been obvious to one of ordinary skill in the art to combine the teachings of Schuster with those of Smith and Pangels since doing so would efficiently detect the movement of the moving body.

Regarding claim 11, Pangels teaches applying a first weight to the first set of data readings, a second weight to the second set of data readings, and third weight to the third set of data readings (col 3, ln 15-60; col 14, ln 1-30; col 18, ln 15-65; col 19, ln 10-60); decreasing the first relative to the second weight and the third weight for data readings indicative of low contrast imagery sensed by the imaging sensor (col 8, ln 45-65; col 14, ln 1-35; col 18, ln 15-65; col 19, ln 10-60); decreasing the second weight relative to the first weight and the third weight for data readings indicative of changes in depth below a predetermined threshold sensed by the imaging sensor (col 6, ln 15-50; col 8, ln 1-45; col 18, ln 15-65; col 19, ln 10-60); and decreasing the third weight relative to the first region and the second weight for data readings indicative of motion of the moving body (col 18, ln 15-65; col 19, ln 10-60).

---- (See Continuation in Supplemental Box) ----

NVIDIA Ex1002 Page 110 of 1081

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.

Continuation of:
Box V.2. Citations and Explanations

Claim 12 lacks an inventive step under PCT Article 33(3) as being obvious over Schuster in view of Smith and further in view of US 2008/0033897 A1 (Lloyd).

Regarding claim 12, Schuster teaches generating at least one first set of data cells representing respective data points in a measurement range of the at least one sensor, the respective data points representing a continuous portion of the environment (para [0045]-[0046], [0050]-[0053]). Neither Schuster nor Smith teaches defining a resolution at the first neural representation corresponding to the at least one first set of data cells based at least in part on a range of a corresponding set of data readings and on a number of data cells in the at least one first set of data cells; determining a scale of the first neural representation corresponding to the at least one first set of data cells based at least in part on the range of the corresponding set of data readings; and generating with the processor a set of graph connections among the at least one first set of data cells, the set of graph connections providing continuity between representations of the respective data points. However, Lloyd teaches defining a resolution at the first neural representation corresponding to the at least one first set of data cells based at least in part on a range of a corresponding set of data readings and on a number of data cells in the at least one first set of data cells (para [0029]-[0031], [0076]-[0079], [0113]-[0116], [0174]-[0177]); determining a scale of the first neural representation corresponding to the at least one first set of data cells based at least in part on the range of the corresponding set of data readings (para [0025], [0073]-[0074], [0192]-[0195], [0200]-[0203]); and generating with the processor a set of graph connections among the at least one first set of data cells, the set of graph connections providing continuity between representations of the respective data points (para [0076]-[0078], [0111], [0145]-[0149], [0196]). It would have been obvious to one of ordinary skill in the art to combine the teachings of Schuster with those of Smith and Lloyd since doing so would provide a visual representation the neural representation of the sensor data.

Claims 1-13 have industrial applicability as defined by PCT Article 33(4) because the subject matter can be made or used in industry.

PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

To: CHRISTOPHER MAX COLICE
1299 PENNSYLVANIA AVENUE, NW
SUITE 700
WASHINGTON, DC 20004-2400

PCT

NOTIFICATION OF TRANSMITTAL OF THE INTERNATIONAL SEARCH REPORT AND THE WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY, OR THE DECLARATION

(PCT Rule 44.1)

Applicant's or agent's file reference NRLA-00401WO	Date of mailing *(day/month/year)* 15 SEP 2015 **FOR FURTHER ACTION** See paragraphs 1 and 4 below
International application No. PCT/US 15/21492	International filing date *(day/month/year)* 19 March 2015 (19.03.2015)
Applicant NEURALA, INC.	

1. ☒ The applicant is hereby notified that the international search report and the written opinion of the International Searching Authority have been established and are transmitted herewith.

   Filing of amendments and statement under Article 19:
   The applicant is entitled, if he so wishes, to amend the claims of the international application (see Rule 46):

   When? The time limit for filing such amendments is normally two months from the date of transmittal of the international search report.

   How? Directly to the International Bureau of WIPO preferably through ePCT or on paper to, 34 chemin des Colombettes 1211 Geneva 20, Switzerland, Facsimile No.: +41 22 338 82 70

   For more detailed instructions, see *PCT Applicant's Guide*, International Phase, paragraphs 9.004 – 9.011.

2. ☐ The applicant is hereby notified that no international search report will be established and that the declaration under Article 17(2)(a) to that effect and the written opinion of the International Searching Authority are transmitted herewith.

3. ☐ **With regard to any protest** against payment of (an) additional fee(s) under Rule 40.2, the applicant is notified that:
   - ☐ the protest together with the decision thereon has been transmitted to the International Bureau together with any request to forward the texts of both the protest and the decision thereon to the designated Offices.
   - ☐ no decision has been made yet on the protest; the applicant will be notified as soon as a decision is made.

4. **Reminders**

   The applicant may **submit comments on an informal basis on the written opinion of the International Searching Authority** to the International Bureau. These comments will be made available to the public after international publication. The International Bureau will send a copy of such comments to all designated Offices unless an international preliminary examination report has been or is to be established.

   Shortly after the expiration of **18 months from the priority date, the international application will be published** by the International Bureau. If the applicant wishes to avoid or postpone publication, a notice of withdrawal of the international application, or of the priority claim, must reach the International Bureau before the completion of the technical preparations for international publication (Rules 90*bis*.1 and 90*bis*.3).

   Within **19 months** from the priority date, but only in respect of some designated Offices, a demand for international preliminary examination must be filed if the applicant wishes to postpone the entry into the national phase **until 30 months** from the priority date (in some Offices even later); otherwise, the applicant must, **within 20 months** from the priority date, perform the prescribed acts for **entry into the national phase** before those designated Offices. In respect of other designated Offices, the time limit of **30 months** (or later) will apply even if no demand is filed within 19 months. For details about the applicable time limits, Office by Office, see www.wipo.int/pct/en/texts/time_limits.html and the *PCT Applicant's Guide*, National Chapters.

   Within **19 months from the priority date, the applicant may request that a supplementary international search be carried out** by a different International Searching Authority that offers this service (Rule 45*bis*.1). The procedure for requesting supplementary international search is described in the *PCT Applicant's Guide*, International Phase, paragraphs 8.006-8.032.

Name and mailing address of the ISA/ Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Lee W. Young PCT Helpdesk: 571-272-4300 Telephone No. PCT OSP: 571-272-7774

Form PCT/ISA/220 (July 2014)

PATENT COOPERATION TREATY

PCT

INTERNATIONAL SEARCH REPORT

(PCT Article 18 and Rules 43 and 44)

Applicant's or agent's file reference NRLA-00401WO	FOR FURTHER ACTION	see Form PCT/ISA/220 as well as, where applicable, item 5 below.
International application No. PCT/US 15/21492	International filing date *(day/month/year)* 19 March 2015 (19.03.2015)	(Earliest) Priority Date *(day/month/year)* 19 March 2014 (19.03.2014)
Applicant NEURALA, INC.		

This international search report has been prepared by this International Searching Authority and is transmitted to the applicant according to Article 18. A copy is being transmitted to the International Bureau.

This international search report consists of a total of 2 sheets.

☐ It is also accompanied by a copy of each prior art document cited in this report.

1. **Basis of the report**

a. With regard to the **language**, the international search was carried out on the basis of:

☒ the international application in the language in which it was filed.

☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

b. ☐ This international search report has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43.6*bis*(a)).

c. ☐ With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, see Box No. I.

2. ☐ **Certain claims were found unsearchable** (see Box No. II).

3. ☐ **Unity of invention is lacking** (see Box No. III).

4. With regard to the **title**,

☒ the text is approved as submitted by the applicant.

☐ the text has been established by this Authority to read as follows:

5. With regard to the **abstract**,

☒ the text is approved as submitted by the applicant.

☐ the text has been established, according to Rule 38.2, by this Authority as it appears in Box No. IV. The applicant may, within one month from the date of mailing of this international search report, submit comments to this Authority.

6. With regard to the **drawings**,

a. the figure of the **drawings** to be published with the abstract is Figure No. 1

☒ as suggested by the applicant.

☐ as selected by this Authority, because the applicant failed to suggest a figure.

☐ as selected by this Authority, because this figure better characterizes the invention.

b. ☐ none of the figures is to be published with the abstract.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US 15/21492

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06K 9/00 (2015.01)
CPC - B25J 9/1697
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
CPC: B25J 9/1697
IPC(8): G06K 9/00 (2015.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC: 382/153 or 382/293; IPC(8): G06K 9/00 (2015.01)
CPC: B25J 9/1697 or G06K 9/00201 or G05D 1/0242 or G05D 1/0246 or G05D 1/0272

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
Patbase; Google Patents; Google Scholar; Google Web
Search Terms Used: Robot, foveated, object, track, recognition, metadata, item, location, position, ID, identify, name, label, layer, view, contour, shroud

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y --- A	US 6,510,244 B2 (CARSON et al.) 13 August 2013 (13.08.2013), col 1, ln 55- col 2, ln 5, col 10, ln 4-18, col 11, ln 49-56, col 12, ln 11-15, col 20, ln 6-10	1-2, 4 --- 3, 5-8, 11-18 --- 9-10, 19-20
X --- Y --- A	US 7,765,029 B2 (FLEISCHER et al.) 27 July 2010 (27.07.2010), col 2, ln 21-28, col 4, ln 13-19, col 4, ln 30-44, col 5, ln 56-67, col 8, ln 64- col 9, ln 10, col 10, ln 12-39, col 11, ln 9-20, col 11, ln 39-51, col 12, ln 17-50, col 13, ln 4-34, col 14, ln 3-22	21-22 --- 3, 5, 13-18, 23 --- 9-10, 19-20
Y	US 2002/0050518 A1 (ROUSTAEI) 02 May 2002 (02.05.2002), para [0158]	6, 16
Y --- A	US 5,136,687 A (EDELMAN et al.) 04 August 1992 (04.08.1992), col 16, ln 33-63	7-8, 17-18 --- 9-10, 19-20
Y	US 2011/0004341 A1 (SARVADEVABHATLA et al.) 06 January 2011 (06.01.2011), para [0050] -[0052], [0054]	11-12
Y	US 2012/0089552 A1 (CHANG et al.) 12 April 2012 (12.04.2012), para [0033]	23

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 August 2015 (10.08.2015)	15 SEP 2015
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer: Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (January 2015)

PATENT COOPERATION TREATY

From the
INTERNATIONAL SEARCHING AUTHORITY

To: CHRISTOPHER MAX COLICE
1299 PENNSYLVANIA AVENUE, NW
SUITE 700
WASHINGTON, DC 20004-2400

PCT

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

(PCT Rule 43*bis*.1)

Date of mailing *(day/month/year)*	15 SEP 2015
Applicant's or agent's file reference NRLA-00401WO	**FOR FURTHER ACTION** See paragraph 2 below

International application No.	International filing date *(day/month/year)*	Priority date *(day/month/year)*
PCT/US 15/21492	19 March 2015 (19.03.2015)	19 March 2014 (19.03.2014)

International Patent Classification (IPC) or both national classification and IPC
IPC(8) - G06K 9/00 (2015.01)
CPC - B25J 9/1697

Applicant NEURALA, INC.

1. This opinion contains indications relating to the following items:

- ☒ Box No. I Basis of the opinion
- ☐ Box No. II Priority
- ☐ Box No. III Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- ☐ Box No. IV Lack of unity of invention
- ☒ Box No. V Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- ☐ Box No. VI Certain documents cited
- ☐ Box No. VII Certain defects in the international application
- ☐ Box No. VIII Certain observations on the international application

2. **FURTHER ACTION**

If a demand for international preliminary examination is made, this opinion will be considered to be a written opinion of the International Preliminary Examining Authority ("IPEA") except that this does not apply where the applicant chooses an Authority other than this one to be the IPEA and the chosen IPEA has notified the International Bureau under Rule 66.1 *bis*(b) that written opinions of this International Searching Authority will not be so considered.

If this opinion is, as provided above, considered to be a written opinion of the IPEA, the applicant is invited to submit to the IPEA a written reply together, where appropriate, with amendments, before the expiration of 3 months from the date of mailing of Form PCT/ISA/220 or before the expiration of 22 months from the priority date, whichever expires later.

For further options, see Form PCT/ISA/220.

Name and mailing address of the ISA/US	Date of completion of this opinion	Authorized officer:
Mail Stop PCT, Attn: ISA/US Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	10 August 2015 (10.08.2015)	Lee W. Young PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/237 (cover sheet) (January 2015)

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 15/21492

Box No. I Basis of this opinion

1. With regard to the **language**, this opinion has been established on the basis of:

   ☒ the international application in the language in which it was filed.

   ☐ a translation of the international application into _____ which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1(b)).

2. ☐ This opinion has been established taking into account the **rectification of an obvious mistake** authorized by or notified to this Authority under Rule 91 (Rule 43*bis*.1(a)).

3. ☐ With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application, this opinion has been established on the basis of a sequence listing:

   a. ☐ forming part of the international application as filed:

      ☐ in the form of an Annex C/ST.25 text file.

      ☐ on paper or in the form of an image file.

   b. ☐ furnished together with the international application under PCT Rule 13*ter*.1(a) for the purposes of international search only in the form of an Annex C/ST.25 text file.

   c. ☐ furnished subsequent to the international filing date for the purposes of international search only:

      ☐ in the form of an Annex C/ST.25 text file (Rule 13*ter*.1(a)).

      ☐ on paper or in the form of an image file (Rule 13*ter*.1(b) and Administrative Instructions, Section 713).

4. ☐ In addition, in the case that more than one version or copy of a sequence listing has been filed or furnished, the required statements that the information in the subsequent or additional copies is identical to that forming part of the application as filed or does not go beyond the application as filed, as appropriate, were furnished.

5. Additional comments:

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 15/21492

Box No. V **Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement**

1. Statement

Novelty (N)	Claims	3, 5-20, 23	YES
	Claims	1-2, 4, 21-22	NO
Inventive step (IS)	Claims	9-10, 19-20	YES
	Claims	1-8, 11-18, 21-23	NO
Industrial applicability (IA)	Claims	1-23	YES
	Claims	None	NO

2. Citations and explanations:

Claims 1-2 and 4 lack novelty under PCT Article 33(2) as being anticipated by US 8,510,244 B2 to Carson et al. (hereinafter, 'Carson').

As per claim 1, Carson discloses a system for automatically locating and identifying an object in an environment (col 1, ln 55 to col 2, ln 5); the system comprising: at least one sensor to acquire sensor data representing of at least a portion of the object (col 1, ln 55 to col 2, ln 5- disclosing an imaging sensor to send sensor data); a spatial attention module, operably coupled to the at least one sensor, to produce a foveated representation of the object based at least in part on the sensor data, to track a position of the object within the environment based at least in part on the foveated representation (col 10, ln 4-18- disclosing a three-dimensional stack for tracking an object); and to select another portion of the environment to be sensed by the at least one sensor based at least in part on the foveated representation of the object (col 11, ln 49-56- disclosing reconfigurable foveated representation based on recognizing an object); and
a semantics module, operably coupled to the spatial attention module, to determine an identity of the object based at least in part on the foveated representation of the object (col 12, ln 11-15; col. 17, ln 9-11).

As per claim 2, Carson discloses the system of claim 1. Carson further discloses wherein the at least one sensor comprises an image sensor to acquire at least one image of the at least a portion of the environment (col 1, ln 59 to col 2, ln 5).

As per claim 4, Carson discloses the system of claim 1. Carson further discloses wherein the spatial attention module configured to select the other portion of the environment such that the object appears at or near a center of the other portion of the environment (col 20, ln 6-10).

Claims 21-22 lack novelty under PCT Article 33(2) as being anticipated by US 7,765,029 B2 to Fleischer et al. (hereinafter, 'Fleischer').

As per claim 21, Fleischer discloses a system for automatically locating and identifying an object in an environment, the system comprising: a plurality of sensors to acquire multi-sensory data representative of the environment (col 2, ln 21-28- disclosing multiple sensors including cameras); a plurality of first spatial attention modules, each first spatial attention module in the plurality of first spatial attention modules operably coupled to a respective sensor in the plurality of sensors and configured to produce a respective representation of the object based on a respective portion of the multi-sensory data (Fig. 3A, col 9, ln 11-15, col 11, ln 9-20- disclosing neuronal and non-neuronal signal processing elements connected to respective sensors for providing respective representation of an object); a plurality of first semantics modules, each first semantics module in the plurality of first spatial attention modules operably coupled to a respective first spatial attention module in the plurality of first spatial attention modules and configured to produce respective identifications of the object based on an output from the respective first spatial attention module (col 11, ln 53-65, col 12, ln 17-37- disclosing 15 neuronal areas each area is a group of neuronal elements used to identify objects based on their respective shape); a second spatial attention module, operably coupled to the plurality of first spatial attention modules, to estimate a location of the object based at least in part on the multisensory representation of the environment (col 4, ln 30-44); and a second semantics module, operably coupled to the plurality of first semantics modules and the second spatial attention module, to identify the at least one object based in part of the respective identifications produced by the plurality of first semantics modules and the location of the object estimated by the second spatial attention module (col 12, ln 17-50- disclosing identifying objects based on their location and spatial context).

As per claim 22, Fleischer discloses the system of claim 21. Fleischer further discloses wherein the second spatial attention module is configured to dynamically allocate the plurality of sensors so as to enhance detection of the at least one object and/or to suppress noise in the multi-sensory representation of the scene (col 13, ln 4-34, col 14, ln 3-22).

---See continuation sheet---

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 15/21492

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.
Continuation of:
Box V.2 Citations and Explanations

Claims 3, 5, and 13-15 lacks an inventive step under PCT Article 33(3) as being obvious over Carson in view of Fleischer.

As per claim 3, Carson discloses the system of claim 2. However, Carson does not disclose at least one actuator, operably coupled to the image sensor, to provide sensor position data representative of an orientation and/or a position of the image sensor, and wherein the spatial attention module is configured to select the other portion of the environment based at least in part on the orientation and/or a position of the image sensor. However, Flesicher discloses at least one actuator, operably coupled to the image sensor (col 4, ln 13-19, col 8, ln 64 to col 9, ln 10- disclosing sensor actuators where sensors can be imaging sensors); to provide sensor position data representative of an orientation and/or a position of the image sensor, and wherein the spatial attention module is configured to select the other portion of the environment based at least in part on the orientation and/or a position of the image sensor (col 10, ln 12-39, col 11, ln 9 -20- disclosing position and orientation information of sensors are used to determine what area of the image should be the focus of the sensor). It would have been obvious to one of ordinary skill in the art to combine the system taught by Carson with the actuators taught by Fleischer, since such would allow for adjusting where the imaging sensor is focused based on received sensor data.

As per claim 5, Carson discloses the system of claim 1. However, Carson does not disclose wherein the spatial attention module comprises: a memory to store the sensor position data representing an orientation and/or a position of the sensor; and an inhibition of return module, operably coupled to the memory, to inhibit repeated selections of a given object based at least in part on the sensor position data stored in the memory. However, Fleischer discloses a memory to store the sensor position data representing an orientation and/or a position of the sensor (col 5, ln 56-67); and an inhibition of return module, operably coupled to the memory, to inhibit repeated selections of a given object based at least in part on the sensor position data stored in the memory (col 12, ln 17-50- disclosing inhibiting the tracking or recognition of objects that should not be in the same position as an object that should be recognized). It would have been obvious to combine the system taught by Carson with the memory taught by Fleischer, since such would allow for storing location of recognized objects for later usage.

As per claim 13, Carson discloses a method of automatically locating and identifying an object in an environment (col 1, ln 55- col 2, ln 5); the method comprising: B acquiring, with the at least one sensor, sensor data representing at least a portion of the object (col 1, ln 55- col 2, ln 5- disclosing an imaging sensor to send sensor data); C. producing a foveated representation of the object based at least in part on the sensor data acquired in B (col 10, ln 4-18- disclosing a three-dimensional stack for tracking an object); D. determining an identity of the object based at least in part on the foveated representation of the object produced in C (col 12, ln 11-15). However, Carson does not disclose A. estimating a position and/or an orientation of at least one sensor with respect to the environment; and E. selecting another portion of the environment to be sensed by the at least one sensor based at least in part on the foveated representation of the object produced in C and the position and/or the orientation estimated in A; F acquiring additional sensor data, with the at least one sensor, in response to selection of the other portion of the environment in D. However, Fleischer does disclose A. estimating a position and/or an orientation of at least one sensor with respect to the environment (col 5, ln 56-67); and E. selecting another portion of the environment to be sensed by the at least one sensor based at least in part on the foveated representation of the object produced in C. and the position and/or the orientation estimated in A (col 13, ln 4-33, col 14, ln 4-22- disclosing keeping an object focused by re-acquiring objects from a foveated object based on a position and orientation of a sensor); F acquiring additional sensor data, with the at least one sensor, in response to selection of the other portion of the environment in D (col 4, ln 7-19). It would have been obvious to one of ordinary skill in the art to combine the method taught by Carson with the position and location taught by Fleischer, since such would allow for adjusting where the imaging sensor is focused based on received sensor data.

As per claim 14, Carson in view of Fleischer discloses the method of claim 13. Fleischer further discloses wherein D comprises selecting the other portion of the environment such that the object appears at or near a center of the other portion of the environment (col 14, ln 4-22).

As per claim 15, Carson in view of Flesicher discloses the method of claim 13. Fleischer further discloses wherein D. comprises inhibiting repeated selections of a given portion of the environment based at least in part on the position estimated in A (col 12, ln 17-50- disclosing inhibiting the tracking or recognition of objects that should not be in the same position as an object that should be recognized).

Claim 6 lacks an inventive step under PCT Article 33(3) as being obvious over Carson in view of US 2002/0050518 A1 (Roustaei).

As per claim 6, Carson discloses the system of claim 1. However, Carson does not disclose wherein the spatial attention module comprises: at least one log-polar transformation module to transform the sensor data into a log-polar representation of the environment so as to provide invariance to translation and/or rotation of the at least one sensor with respect to the object and/or so as to reduce processing time. However, Roustaei discloses at least one log-polar transform1ation module to transform the sensor data into a log-polar representation of the environment so as to provide invariance to translation and/or rotation of the at least one sensor with respect to the object and/or so as to reduce processing time (para [0158]). It would have been obvious to one of ordinary skill in the art to combine the system taught by Carson with the log-polar transformation taught by Roustaei since such would allow for normalizing sensor data.

---See next page---

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 15/21492

Supplemental Box

In case **the space in any of the preceding boxes is not sufficient.**
Continuation of:
Box V.2 Citations and Explanations

Claims 7-8 lack an inventive step under PCT Article 33(3) as being obvious over Carson in view of US 5,136,687 A to Edelman et al. (hereinafter, 'Edelman').

As per claim 7, Carson discloses the system of claim 1. However, Carson does not disclose wherein the spatial attention module comprises: a segmentation module to generate at least one contour representation of the object based at least in part on the sensor data. However, Edelman does disclose wherein the spatial attention module comprises: a segmentation module to generate at least one contour representation of the object based at least in part on the sensor data (col 16, ln 33-63- disclosing object recognition based upon recognizing contours of objects). It would have been obvious to one of ordinary skill in the art to combine the system taught by Carson with the contours taught by Edelman, since such would allow for quickly recognizing objects based on their contours rather than the object in its entirety.

As per claim 8, Carson in view of Edelman discloses the system of claim 7. Carson further discloses wherein the spatial attention module further comprises: a figure/ground segregation module, operably coupled to the segmentation module, to determine at least one spatial shroud fitting a form of the object based at least in part on the at least one contour representation of the object (col 1, ln 59- col 2, ln 5- disclosing detecting an object based on general shape of the object).

Claims 11-12 lack an inventive step under PCT Article 33(3) as being obvious over Carson in view of US 2011/0004341 A1 to Sarvadevabhatla et al. (hereinafter, 'Sarvadevabhatla').

As per claim 11, Carson discloses the system of claim 1. However, Carson does not disclose wherein the semantics module is configured to learn the identity of the object based at least in part of the location of the object. However, Sarvadevabhatla disclose wherein the semantics module is configured to learn the identity of the object based at least in part of the location of the object (para [0050]-[0052]- disclosing knowing previous location and matching with current location to learn the identity of an object). It would have been obvious to one of ordinary skill in the art to combine the system taught by Carson with the learning the identity taught by Sarvadevabhatla since such would allow for quickly identifying an object.

As per claim 12, Carson discloses the system of claim 1. However, Carson does not disclose a teaching module, operably coupled to the at least one sensor and to the semantic system, to provide a label for the object, and wherein the semantics module is configured to assign the label to the object. However, Sarvadevabhatla discloses a teaching module, operably coupled to the at least one sensor and to the semantic system, to provide a label for the object, and wherein the semantics module is configured to assign the label to the object (para [0050]-[0052], [0054]- disclosing storing identities of an object in an attention entity module for later assigning an identity to an object). It would have been obvious to one of ordinary skill in the art to combine the system taught by Carson with the learning the identity taught by Sarvadevabhatla since such would allow for quickly identifying an object.

Claim 23 lacks an inventive step under PCT Article 33(3) as being obvious over Fleischer in view of US 2012/0089552 A1 to Chang et al. (hereinafter, 'Chang').

As per claim 23, Fleischer discloses the system of claim 21. However, Fleischer does not disclose wherein the second semantics module is configured to generate scene metadata representative of an identity of the at least one object and/or the at least one change to the object based at least in part on the multi-sensory representation of the environment. However, Chang discloses wherein the second semantics module is configured to generate scene metadata representative of an identity of the at least one object and/or the at least one change to the object based at least in part on the multi-sensory representation of the environment (para [0033]). It would have been obvious to one of ordinary skill in the art to combine the system taught by Fleischer with the metadata taught by Chang, since such would allow for utilizing data retrieved from the scene to make object recognition more efficient.

Claim 16 lacks an inventive step under PCT Article 33(3) as being obvious over Carson in view of Fleischer and further in view of Roustaei.

As per claim 16, Carson in view of Fleischer discloses the method of claim 13. However, Carson in view of Flesicher does not disclose wherein D. comprises transforming the sensor data into a log-polar representation of the environment so as to provide invariance to translation and/or rotation and/or so as to reduce processing time. However, Roustaei does disclose wherein D. comprises transforming the sensor data into a log-polar representation of the environment so as to provide invariance to translation and/or rotation and/or so as to reduce processing time (para [0158]). It would have been obvious to one of ordinary skill in the art to combine the method taught by Carson and Fleischer with the log-polar transformation taught by Roustaei since such would allow for normalizing sensor data.

---See continuation sheet---

WRITTEN OPINION OF THE
INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US 15/21492

Supplemental Box

In case the space in any of the preceding boxes is not sufficient.
Continuation of:
Box V.2 Citations and Explanations

Claims 17-18 lack an inventive step under PCT Article 33(3) as being obvious over Carson in view of Fleischer and further in view of Edelman.

As per claim 17, Carson in view of Fleischer discloses the method of claim 13. Fleischer further discloses wherein: A. comprises acquiring a plurality of images (col 9, ln 48-65). However, Carson in view of Fleischer does not disclose D. comprises generating at least one contour representation of the object based at least in part on the at least one image. However, Edelman does disclose D. comprises generating at least one contour representation of the object based at least in part on the at least one image (col 16, ln 33-63- disclosing object recognition based upon recognizing contours of objects). It would have been obvious to one of ordinary skill in the art to combine the system taught by Carson with the contours taught by Edelman, since such would allow for quickly recognizing objects based on their contours rather than the object in its entirety.

As per claim 18, Carson in view of Fleischer and Edelman disclose the method of claim 17. Carson further discloses wherein D further comprises determining at least one spatial shroud fitting a form of the object based at least in part on the at least one contour representation of the object (col 1, ln 59- col 2, ln 5- disclosing detecting an object based on general shape of the object).

Claims 9-10 and 19-20 meet the criteria set forth under PCT Article 33(2)-(3) because the prior art fails to disclose or suggest the particular combination of elements recited in claims 9-10 and 19-20.

The prior art is exemplified by Carson, Fleischer, and Edelman.

As per claim 9, Carson and Edelman discloses the system of claim 8. However, Carson and Edelman does not disclose wherein the sensor data comprises a plurality of images and the semantics module comprises: a view layer to group views of the object in the plurality of images based at least in part on the at least one spatial shroud; an object layer, operably coupled to the view layer, to map the views of the object to an object node associated with the object; and a name layer, opernbly coupled to the object layer, to classify the object based at least in part on the object node. Further, the prior art, whether taken alone or in combination, fails to disclose or suggest the particular arrangement of these parts as required by the claim.

Claim 10 is dependent on claim 9 and therefore meet the criteria for substantially the same reasons.

As per claim 19, Carson in view of Fleischer and Edelman discloses the method of claim 18. However, Carson in view of Fleischer and Edelman does not disclose wherein E. comprises: E1 grouping views of the object in the plurality of images based at least in part on the at least one spatial shroud; E2. mapping the views of the object to an object node associated with the object; and E3. classifying the object based at least in part on the object node. Further, the prior art, whether taken alone or in combination, fails to disclose or suggest the particular arrangement of these parts as required by the claim.

Claim 20 is dependent on claim 19 and therefore meet the criteria for substantially the same reasons.

Claims 1-23 have industrial applicability as defined by PCT Article 33(4) because the subject matter can be made or used in industry.

Electronic Acknowledgement Receipt

EFS ID:	31169320
Application Number:	15808201
International Application Number:	
Confirmation Number:	1077
Title of Invention:	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD
First Named Inventor/Applicant Name:	Anatoli Gorchetchnikov
Customer Number:	58249
Filer:	Nisha Somnath/Donna Doyle
Filer Authorized By:	Nisha Somnath
Attorney Docket Number:	NRLA-001/03US 320730-2040
Receipt Date:	08-DEC-2017
Filing Date:	
Time Stamp:	19:56:20
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Non Patent Literature	NRLA00601WO_ISRandWO_2016-02-23.pdf	12868984 35611edbc037831918919a65bfe128c8fa6b5cf1	no	11

Warnings:

Information:					
2	Non Patent Literature	NRLA00901WO_ISRandWO_2017-07-06.pdf	2087847 eb68caefb3b6a84a5b749ac814c592ad133a4f1d	no	12
Warnings:					
Information:					
3	Non Patent Literature	NRLA00301WO_ISRandWO_2014-11-26.pdf	2346687 1765729620413a990c87daeb0c07d3c88d32ab59	no	14
Warnings:					
Information:					
4	Non Patent Literature	NRLA00401WO_ISRandWO_2015-09-15.pdf	3177530 2e02e1cd4fa5b2dfdab29bc2f14c8ef6599eb46e	no	9
Warnings:					
Information:					
5	Non Patent Literature	Itti_ComputationalModellingOfVisualAttention.pdf	744927 e49b8301af88e47e852a88ada06ea17771694fce	no	10
Warnings:					
Information:					
6	Non Patent Literature	Itti_AModelOfSaliency-BasedVisualAttentionForRapidSceneAnalysis.pdf	15378254 d185549f8710aed5a91576853c77a22ff18a448c	no	6
Warnings:					
Information:					
7	Non Patent Literature	Jarret_WhatIsTheBestMulti-stageArchitectureForObjectRecognition.pdf	859381 1f6a5330b40b7739ffc6b2a7937a2a79d7475076	no	8
Warnings:					
Information:					
8	Non Patent Literature	Kipfer.pdf	3102572 9e92d6d0b225fa3933e3be89712b037ce57732fc	no	9
Warnings:					
Information:					

9	Non Patent Literature	Kolb.pdf	3476161 2e2b29af88bb794dd97c5c7ceecf336ac877a737	no	9
Warnings:					
Information:					
10	Non Patent Literature	Kompella.pdf	1694931 0ea0ed09391f3bc4eecc00dc6fcf3e90616595c8	no	39
Warnings:					
Information:					
11	Non Patent Literature	Kowler_EyeMovementsThePast25years.pdf	2258122 f8089553f31a3cf18d9179845b0fd4e09ecc30aa	no	27
Warnings:					
Information:					
12	Non Patent Literature	Larochelle_LearningToCombineFovealGlimpsesWithAThird-orderBoltzmannMachine.pdf	271654 22332bc511e73ddc9d406ff63165a104b2fee2a7	no	9
Warnings:					
Information:					
13	Non Patent Literature	Lecun_ConvolutionalNetworksAndApplicationsInVision.pdf	280469 1e92d08f061167c10679838205a6aec1c170656c	no	4
Warnings:					
Information:					
14	Non Patent Literature	Lee_LearningThePartsOfObjectsByNon-negativeMatrixFactorization.pdf	192787 00b0a58a02c270e81617fb9b2b9788cb38131960	no	4
Warnings:					
Information:					
15	Non Patent Literature	Lee_UnsupervisedLearningByConvexAndConicCoding.pdf	350765 8074ef162dee25ccdad620edd90f5b010687a272	no	7
Warnings:					
Information:					

16	Non Patent Literature	Legenstein_ReinforcementLearningOnSlowFeaturesOfHigh-dimensionalInputStreams.pdf	632439 ed79b60ec34bb14a8046dce066cb347040218ffe	no	13
Warnings:					
Information:					
17	Non Patent Literature	Leveille_LearningInADistributedSoftwareArchitectureForLarge-scaleNeuralModeling.pdf	318111 f40542f7512b782ab12509d3a230c2c2cca2e972	no	8
Warnings:					
Information:					
18	Non Patent Literature	Livitz_AdaptiveBrain-likeSystems.pdf	459911 a7f8963c9c7e1b65e499f57e2e7f39bcddcb2fb7	no	3
Warnings:					
Information:					
19	Non Patent Literature	Livitz_Visually-GuidedAdaptiveRobotViGuAR.pdf	751575 b7ccc523524720350889982b9d84adb1be57ab3d	no	9
Warnings:					
Information:					
20	Non Patent Literature	Lowe_DistinctiveImageFeaturesFromScale-InvariantKeypoints.pdf	691008 9307efc4ed985aa1fa184b6610a4dd5354e049c6	no	28
Warnings:					
Information:					
21	Non Patent Literature	Lu_ModelingMechanismsOfPerceptualLearningWithAugmentedHebbianRe-weighting.pdf	1757481 73b06d1f90c4b89b5cc6b708b1da0feb6608a4ae	no	16
Warnings:					
Information:					
22	Non Patent Literature	Mahadevan_Protovaluefunctionsdevelopmentalreinforcementlearning.pdf	1454916 7c80400e540e44ea9d73f5491cc0a93d02c01bf6	no	8
Warnings:					
Information:					

23	Non Patent Literature	Meuth.pdf	212505 a52e8273c3b7326282e8a2d9049790e2e3c6b301	no	5
Warnings:					
Information:					
24	Non Patent Literature	Mishkin.pdf	22623734 b6ce9a6248c421fc97ac736bbebbeadc87396b2f	no	21
Warnings:					
Information:					
25	Non Patent Literature	Mnih_HumanLevelControlThroughDeepReinforcementLearning.pdf	3731999 10d1b3c739006f8f51561c3b06f575a8fd4d9be8	no	13
Warnings:					
Information:					
26	Non Patent Literature	Moore_1993.pdf	15286651 2a4dd2010d7f1847dea1466ebdc2f0b4854d8c97	no	28
Warnings:					
Information:					
27	Non Patent Literature	Najemnik_SimpleSummationRuleForOptimalFixationSelectionInVisualSearch.pdf	955766 4cdcb4e629c9d6a9aa0452613995df6e65eeb080	no	9
Warnings:					
Information:					
28	Non Patent Literature	Oh.pdf	339781 fde84441fdde53b1d4aa2d26dafbbb73bffd764d	no	4
Warnings:					
Information:					
29	Non Patent Literature	Oja.pdf	5713181 5f40d066e33bcde5ebbd3800259e7f7f7542fa14	no	7
Warnings:					
Information:					

30	Non Patent Literature	NRLA00301EPSearchReport2017-07-04.pdf	1107499 4a19b7d020be3ae4bb458eca2cf57f8c6440bca9	no	13
Warnings:					
Information:					
31	Non Patent Literature	Raijmakers.pdf	18862003 c76d69573beb5526ba360d8a780b59a7b9504409	no	21
Warnings:					
Information:					
32	Non Patent Literature	Ranzato_UnsupervisedLearningOfInvariantFeatureHierarchiesWithApplicationsToObjectRecognition.pdf	503379 6a7ab11a215e0127b4b177dcd4bff5b5ad8f3f29	no	9
Warnings:					
Information:					
33	Non Patent Literature	Raudies_LearningtoNavigateinaVirtualWorldUsingOpticFlowandStereoDisparitySignals.pdf	867666 9c3904140dd80862c355f705bcad7d966cb5d388	no	15
Warnings:					
Information:					
34	Non Patent Literature	Riesenhuber_Hierarchiacal.pdf	274460 d7da90708bd7a25cbe95c2763ad74fb0944996b4	no	7
Warnings:					
Information:					
35	Non Patent Literature	Riesenhuber.pdf	216704 7096b119510656853ad4f63b49fc1a57e1708246	no	6
Warnings:					
Information:					
36	Non Patent Literature	Rolfes.pdf	3238577 3b7716afd3d32b32a97b02e6517da0e8b3e5b766	no	7
Warnings:					
Information:					

37	Non Patent Literature	Rublee-ORBAnEfficientAlternativeToSIFTorSURF.pdf	1082548 9b6764ffb1620cb4d08824615e23da8bba6a8668	no	8
Warnings:					
Information:					
38	Non Patent Literature	Ruesch.pdf	1137888 802b75ba5388f770c77be30586cf7b6bc82951dc	no	6
Warnings:					
Information:					
39	Non Patent Literature	Rumelhart_LearningInternalRepresentationsByErrorPropagation.pdf	1688009 ab7a4ba4ae2150e00186d1513c564ac6e0f8ee9d	no	45
Warnings:					
Information:					
40	Non Patent Literature	Rumpf.pdf	1243154 92f86108a6c5e1a2954e0485ffb5de2543c7405b	no	46
Warnings:					
Information:					
41	Non Patent Literature	Salakhutdinov-DeepBoltzmannMachines.pdf	522102 05cd9eb75fd62c14f1620732766032fbdc185490	no	40
Warnings:					
Information:					
42	Non Patent Literature	Schaul_PrioritizedExperienceReplay.pdf	4235511 bd4a7c1edc738f77022f4d12d0becadd92184920	no	21
Warnings:					
Information:					
43	Non Patent Literature	Schmidhuber-FormalTheoryOfCreativityFunandIntrinsicMotivation.pdf	464457 3d52d68d1f2f608db32549f040cfd7075f951d97	no	18
Warnings:					
Information:					

44	Non Patent Literature	SchmidhuberCuriousModelBuildingControlSystems.pdf	735108 d811319e8605cfdfa11213cb9de727e2f27b5466	no	6
Warnings:					
Information:					
45	Non Patent Literature	Seibert.pdf	13722497 2e99709587724d7b90168bff9c048675e55a20c0	no	18
Warnings:					
Information:					
46	Non Patent Literature	Sherbakov_ComputationalPrinciplesForAnAutonomousActiveVisionSystem.pdf	5347364 752d9a2227d2f989687f54c4ea1e2fbf40ab8f1b	no	194
Warnings:					
Information:					
47	Non Patent Literature	SherbakovCogeyeYouTube.pdf	116888 6544fff61fc6b018d29663519012a4599a5fa8e6	no	1
Warnings:					
Information:					
48	Non Patent Literature	SherbakovCogeyeSystemDiagram.pdf	957468 e57f2887c3939a4c3034dc7028fe73f5768d2482	no	1
Warnings:					
Information:					
49	Non Patent Literature	Sherbakov_CogEyeAnOnlineActiveVisionSystem.pdf	29500 2905addbc424e6524e2a83b89d6251be9b082b31	no	2
Warnings:					
Information:					
50	Non Patent Literature	Sherbakov_Acomputationalmodeloftheroleofeye-movements.pdf	175794 3d3beff3cdc8bf219426751987f21f7c74b2726	no	2
Warnings:					
Information:					

51	Non Patent Literature	Smolensky_1986.pdf	3617627 a5651359364be156850194eb6bacfd98a40d3c4e	no	88
Warnings:					
Information:					
52	Non Patent Literature	Snider_.pdf	1623491 6251a15257c12738f74a717e3c3988e7de85451a	no	9
Warnings:					
Information:					
53	Non Patent Literature	Spratling_PredictiveCodingasaModelofBiasedCompetitionInVisualAttention.pdf	280286 dc9a7071ce9a9eb68e64bf27e6b7c238c8504b93	no	18
Warnings:					
Information:					
54	Non Patent Literature	Spratling_UnsupervisedLearningofGenerative.pdf	1569370 8428b07f1236f697a9c8010efd8876a8d9f534d1	no	44
Warnings:					
Information:					
55	Non Patent Literature	Spratling_UnsupervisedLearningofOverlapping.pdf	695599 86e88afd4ba01a45b26a9db6aa1aa98d2825b187	no	20
Warnings:					
Information:					
56	Non Patent Literature	Sprekeler_OnTheRelationofSlowFeatureAnalysisAndLaplacianEigenmaps.pdf	169723 514344af3f0e66e6a25fb9057b82ba96dabae06d	no	16
Warnings:					
Information:					
57	Non Patent Literature	Sutton.pdf	110547 0608f6dbb4de997155387cce4ffe995b54ed5027	no	10
Warnings:					
Information:					

58	Non Patent Literature	Tong.pdf	2330572 db7779aa7a9d39582f93b0510aa4399db4cbfd15	no	12
Warnings:					
Information:					
59	Non Patent Literature	Torralba.pdf	3951639 9498327c030f7e7e9fa87c72b2611b4e2a930931	no	21
Warnings:					
Information:					
60	Non Patent Literature	VanHasselt_DeepReinforcementLearningWithDoubleq-learning.pdf	629364 8e3edfa6e83e2b6c6f7707c4893aa0596a7582ea	no	7
Warnings:					
Information:					
			Total Files Size (in bytes):	175526923	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.
<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.
<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/08a (07-09)
Approved for use through 07/31/2016. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT

(Use as many sheets as necessary)

Complete if Known	
Application Number	15/808,201
Filing Date	11-09-2017
First Named Inventor	Anatoli Gorchetchnikov
Art Unit	2611
Examiner Name	Not Yet Assigned
Attorney Docket Number	NRLA-001/03US 320730-2040

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code2 (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	1.	5,136,687	08-04-1992	Edelman et al.	
	2.	5,172,253	12-15-1992	Lynne	
	3.	5,388,206	02-07-1995	Poulton et al.	
	4.	6,336,051	01-01-2002	Pangels et al.	
	5.	6,647,508	11-11-2003	Zalewski et al.	
	6.	7,765,029	07-27-2010	Fleischer et al.	
	7.	7,861,060	12-01-2010	Nickolls et al.	
	8.	7,873,650	01-18-2011	Chapman et al.	
	9.	8,392,346	03-05-2013	Ueda et al.	
	10.	8,510,244	08-13-2013	Carson et al.	
	11.	8,648,867	02-11-2014	Gorchetchnikov et al.	
	12.	9,031,692	05-12-2015	Zhu	
	13.	9,177,246	11-03-2015	Buibas et al.	
	14.	9,626,566	04-18-2017	Versace et al.	
	15.	2002/0046271	04-18-2002	Huang	
	16.	2002/0050518	05-02-2002	Roustaei	
	17.	2003/0078754	04-24-2003	Hamza	
	18.	2004/0015334	01-22-2004	Ditlow et al.	
	19.	2005/0166042	07-28-2005	Evans	
	20.	2006/0184273	08-17-2006	Sawada et al.	
	21.	2007/0052713	03-08-2007	Chung et al.	
	22.	2007/0198222	08-23-2007	Schuster et al.	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT

(Use as many sheets as necessary)

Complete if Known	
Application Number	15/808,201
Filing Date	11-09-2017
First Named Inventor	Anatoli Gorchetchnikov
Art Unit	2611
Examiner Name	Not Yet Assigned
Attorney Docket Number	NRLA-001/03US 320730-2040

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code2 (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	23.	2007/0279429	12-06-2007	Ganzer	
	24.	2008/0258880	10-23-2008	Smith et al.	
	25.	2008/0033897	02-07-2008	Lloyd	
	26.	2008/0066065	03-13-2008	Kim et al.	
	27.	2009/0089030	04-02-2009	Sturrock et al.	
	28.	2010/0048242	02-25-2010	Rhoads et al.	
	29.	2011/0004341	01-06-2011	Sarvadevabhatla et al.	
	30.	2011/0173015	07-14-2011	Chapman et al.	
	31.	2012/0072215	03-22-2012	Yu et al.	
	32.	2012/0089552	04-12-2012	Chang et al.	
	33.	2012/0197596	08-02-2012	Comi	
	34.	2012/0316786	12-13-2012	Liu et al.	
	35.	2014/0052679	11-20-2014	Sinyavskiy et al.	
	36.	2014/0089232	03-27-2014	Buibas et al.	
	37.	2014/0019392	01-16-2014	Buibas et al.	
	38.	2014/0032461	01-30-2014	Weng	
	39.	2015/0127149	05-07-2015	Sinyavskiy et al.	
	40.	2015/0134232	05-14-2015	Robinson	
	41.	2016/0075017	03-17-2016	Laurent et al.	
	42.	2016/0082597	03-24-2016	Gorchetchnikov et al.	
	43.	2016/0096270	04-07-2016	Ibarz Gabardos et al.	
	44.	2016/0198000	07-07-2016	Gorchetchnikov et al.	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT

(Use as many sheets as necessary)

Complete if Known	
Application Number	15/808,201
Filing Date	11-09-2017
First Named Inventor	Anatoli Gorchetchnikov
Art Unit	2611
Examiner Name	Not Yet Assigned
Attorney Docket Number	NRLA-001/03US 320730-2040

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code2 (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	45.	2017/0024877	01-26-2017	Versace et al.	
	46.	2017/0076194	03-16-2017	Versace et al.	
	47.	2017/0193298	07-06-2017	Versace et al.	

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]-Number[4]-Kind Code[5] (*if known*)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages Or Relevant Figures Appear	T[6]
	48.	EP 1 224 622 B1	07-24-2002	Sun Microsystems, Inc.		
	49.	WO 2014/190208	11-27-2014	Neurala, Inc.		
	50.	WO 2014/204615	12-24-2014	Neurala, Inc.		
	51.	WO 2015/143173	09-24-2015	Neurala, Inc.		
	52.	WO 2016/014137	01-28-2016	Neurala, Inc.		

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	53.	Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., & Ogden, J. M. (1984). Pyramid methods in image processing. RCA engineer, 29(6), 33-41.	
	54.	Aggarwal, Charu C, Hinneburg, Alexander, and Keim, Daniel A. On the surprising behavior of distance metrics in high dimensional space. Springer, 2001.	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT (Use as many sheets as necessary)	Complete if Known	
	Application Number	15/808,201
	Filing Date	11-09-2017
	First Named Inventor	Anatoli Gorchetchnikov
	Art Unit	2611
	Examiner Name	Not Yet Assigned
	Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	55.	Ames, H, Versace, M., Gorchetchnikov, A., Chandler, B., Livitz, G., Léveillé, J., Mingolla, E., Carter, D., Abdalla, H., and Snider, G. (2012) Persuading computers to act more like brains. In Advances in Neuromorphic Memristor Science and Applications, Kozma, R.Pino,R., and Pazienza, G. (eds), Springer Verlag.	
	56.	Ames, H. Mingolla, E., Sohail, A., Chandler, B., Gorchetchnikov, A., Léveillé, J., Livitz, G. and Versace, M. (2012) The Animat. IEEE Pulse, February 2012, 3(1), 47-50.	
	57.	Artificial Intelligence As a Service. Invited talk, Defrag, Broomfield, CO, Nov 4-6 (2013).	
	58.	Aryananda, L. (2006). Attending to learn and learning to attend for a social robot. Humanoids 06, pp. 618-623.	
	59.	Baraldi, A. and Alpaydin, E. (1998). Simplified ART: A new class of ART algorithms. International Computer Science Institute, Berkeley, CA, TR-98-004, 1998.	
	60.	Baraldi, A. and Alpaydin, E. (2002). Constructive feedforward ART clustering networks – Part I. IEEE Transactions on Neural Networks 13(3), 645–661.	
	61.	Baraldi, A. and Parmiggiani, F. (1997). Fuzzy combination of Kohonen's and ART neural network models to detect statistical regularities in a random sequence of multi-valued input patterns. In International Conference on Neural Networks, IEEE.	
	62.	Baraldi, Andrea and Alpaydin, Ethem. Constructive feedforward ART clustering networks—part II. IEEE Transactions on Neural Networks, 13(3):662–677, May 2002. ISSN 1045-9227. doi: 10.1109/tnn.2002.1000131. URL http://dx.doi.org/10.1109/tnn.2002.1000131.	
	63.	Bengio, Y., Courville, A., & Vincent, P. Representation learning: A review and new perspectives, IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 35 Issue 8, August 2013, Pages 1798-1828 .	

Examiner Signature:		**Date Considered**	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT (Use as many sheets as necessary)	Complete if Known	
	Application Number	15/808,201
	Filing Date	11-09-2017
	First Named Inventor	Anatoli Gorchetchnikov
	Art Unit	2611
	Examiner Name	Not Yet Assigned
	Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	64.	Berenson, D. et al., A robot path planning framework that learns from experience, 2012 International Conference on Robotics and Automation, 2012, 9 pages [retrieved from the internet] URL:http://users.wpi.edu/-dberenson/lightning.pdf.	
	65.	Bernhard, F., and Keriven, R. (2005). Spiking Neurons on GPUs. Tech. Rep. 05-15, Ecole Nationale des Ponts et Chauss´es, 8 pages.	
	66.	Besl, P. J., & Jain, R. C. (1985). Three-dimensional object recognition. ACM Computing Surveys (CSUR), 17(1), 75-145.	
	67.	Bohn, C.-A. Kohonen. (1998). Feature Mapping Through Graphics Hardware. In Proceedings of 3rd Int. Conference on Computational Intelligence and Neurosciences, 4 pages.	
	68.	Bradski, G., & Grossberg, S. (1995). Fast-learning VIEWNET architectures for recognizing three-dimensional objects from multiple two-dimensional views. Neural Networks, 8 (7-8), 1053-1080.	
	69.	Brain-inspired computing. Invited keynote address, Bionetics 2010, Boston, MA, USA.	
	70.	Canny, J.A. (1986). Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698.	
	71.	Carpenter, G.A. and Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing 37, 54–115.	
	72.	Carpenter, G.A., and Grossberg, S. (1995). Adaptive resonance theory (ART). In M. Arbib (Ed.), The handbook of brain theory and neural networks. (pp. 79–82). Cambridge, M.A.: MIT press.	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT (Use as many sheets as necessary)	Complete if Known	
	Application Number	15/808,201
	Filing Date	11-09-2017
	First Named Inventor	Anatoli Gorchetchnikov
	Art Unit	2611
	Examiner Name	Not Yet Assigned
	Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	73.	Carpenter, G.A., Grossberg, S. and Rosen, D.B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. Neural Networks 4, 759–771.	
	74.	Carpenter, Gail A and Grossberg, Stephen. The art of adaptive pattern recognition by a self-organizing neural network. Computer, 21(3):77–88, 1988.	
	75.	Coifman, R.R. and Maggioni, M. Diffusion wavelets. Applied and Computational Harmonic Analysis, 21(1):53–94, 2006.	
	76.	Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S.W. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proceedings of the National Academy of Sciences of the United States of America, 102(21):7426, 2005.	
	77.	Davis, C. E. 2005. Graphic Processing Unit Computation of Neural Networks. Master's thesis, University of New Mexico, Albuquerque, NM, 121 pages.	
	78.	Dosher, B.A., and Lu, Z.L. (2010). Mechanisms of perceptual attention in precuing of location. Vision Res., 40(10–12). 1269–1292.	
	79.	Ellias, S. A., and GROSSBERG, S. (1975). Pattern formation, contrast control and oscillations in the short term memory of shunting on-center off-surround networks. Biol Cybern 20, pp. 69–98.	
	80.	Extended European Search Report and Written Opinion mailed June 1, 2017 from European Application No. 14813864.7, 10 pages (Atty. Docket No. NRLA-002/01EP).	
	81.	Extended European Search Report and Written Opinion mailed October 12, 2017 from European Application No. 14800348.6, 12 pages (Atty. Docket No. NRLA-003/01EP).	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT (Use as many sheets as necessary)	Complete if Known	
	Application Number	15/808,201
	Filing Date	11-09-2017
	First Named Inventor	Anatoli Gorchetchnikov
	Art Unit	2611
	Examiner Name	Not Yet Assigned
	Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	82.	Extended European Search Report and Written Opinion October 23, 2017 from European Application No. 15765396.5, 8 pages (Atty. Docket No. NRLA-004/01EP).	
	83.	Fazl, A., Grossberg, S., and Mingolla, E. (2009). View-invariant object category learning, recognition, and search: How spatial and object attention are coordinated using surface-based attentional shrouds. Cognitive Psychology 58, 1–48.	
	84.	Földiák, P. (1990). Forming sparse representations by local anti-Hebbian learning, Biological Cybernetics, vol. 64, pp. 165-170.	
	85.	Friston K., Adams R., Perrinet L., & Breakspear M. (2012). Perceptions as hypotheses: saccades as experiments. Frontiers in Psychology, 3 (151), 1-20.	
	86.	Galbraith, B.V, Guenther, F.H., and Versace, M. (2015) A neural network-based exploratory learning and motor planning system for co-robots.Frontiers in Neuroscience, in press.	
	87.	George, D. and Hawkins, J. (2009). Towards a mathematical theory of cortical micro-circuits. PLoS Computational Biology 5(10), 1–26.	
	88.	Georgeii, J., and Westermann, R. (2005). Mass-spring systems on the GPU. Simulation Modelling Practice and Theory 13, pp. 693–702.	
	89.	Gorchetchnikov A., Hasselmo M.E. (2005). A biophysical implementation of a bidirectional graph search algorithm to solve multiple goal navigation tasks. Connection Science, 17(1-2), pp. 145-166.	
	90.	Gorchetchnikov A., Hasselmo M.E. (2005). A simple rule for spike-timing-dependent plasticity: local influence of AHP current. Neurocomputing, 65-66, pp. 885-890.	
	91.	Gorchetchnikov A., Versace, M., Hasselmo M.E. (2005). A Model of STDP Based on Spatially and Temporally Local Information: Derivation and Combination with Gated Decay. Neural Networks, 18, pp. 458-466.	

Examiner Signature:		**Date Considered**	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT (Use as many sheets as necessary)	Complete if Known	
	Application Number	15/808,201
	Filing Date	11-09-2017
	First Named Inventor	Anatoli Gorchetchnikov
	Art Unit	2611
	Examiner Name	Not Yet Assigned
	Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	92.	Gorchetchnikov A., Versace, M., Hasselmo M.E. (2005). Spatially and temporally local spike-timing-dependent plasticity rule. In: Proceedings of the International Joint Conference on Neural Networks, number 1568 in IEEE CD-ROM Catalog Number: 05CH37662C, pp. 390-396.	
	93.	Gorchetchnikov A. An Approach to a Biologically Realistic Simulation of Natural Memory. Master's thesis, Middle Tennessee State University, Murfreesboro, TN, 70 pages.	
	94.	Grossberg, S. (1973). Contour enhancement, short-term memory, and constancies in reverberating neural networks. Studies in Applied Mathematics 52, 213–257.	
	95.	Grossberg, S., and Huang, T.R. (2009). ARTSCENE: A neural system for natural scene classification. Journal of Vision, 9 (4), 6.1–19. doi:10.1167/9.4.6	
	96.	Grossberg, S., and Versace, M. (2008) Spikes, synchrony, and attentive learning by laminar thalamocortical circuits. Brain Research, 1218C, 278-312 [Authors listed alphabetically].	
	97.	Hagen, T.R., Hjelmervik, J., Lie, K.-A., Natvig, J. and Ofstad Henriksen, M. (2005). Visual simulation of shallow-water waves. Simulation Modelling Practice and Theory 13, pp. 716–726.	
	98.	Hasselt, Hado Van. Double q-learning. In Advances in Neural Information Processing Systems, pp. 2613–2621, 2010.	
	99.	Hinton, G. E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18, 1527–1554.	
	100.	Hodgkin, A.L., and Huxley, A.F. (1952). Quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol 117, pp. 500–544.	
	101.	Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. In Proc Natl Acad Sci USA, vol. 79, pp. 2554–2558.	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT (Use as many sheets as necessary)	Complete if Known	
	Application Number	15/808,201
	Filing Date	11-09-2017
	First Named Inventor	Anatoli Gorchetchnikov
	Art Unit	2611
	Examiner Name	Not Yet Assigned
	Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	102.	Ilie, A. (2002). Optical character recognition on graphics hardware. Tech. Rep. integrative paper, UNCCH, Department of Computer Science, 9 pages.	
	103.	International Preliminary Report on Patentability in related PCT Application No. PCT/US2014/039162 filed 22 May 2014, mailed 24 November 2015, 7 pages.	
	104.	International Preliminary Report on Patentability in related PCT Application No. PCT/US2014/039239 filed 22 May 2014, mailed 24 November 2015, 8 pages.	
	105.	International Preliminary Report on Patentability mailed November 8, 2016 from International Application No. PCT/US2015/029438, 7 pages (Atty. Docket No. NRLA-006/01WO).	
	106.	International Search Report and Written Opinion mailed February 18, 2015 from International Application No. PCT/US2014/039162, 12 pages (Atty. Docket No. NRLA-002/01WO).	
	107.	International Search Report and Written Opinion mailed February 23, 2016 from International Application No. PCT/US2015/029438, 11 pages (Atty. Docket No. NRLA-006/01WO).	
	108.	International Search Report and Written Opinion mailed July 6, 2017 from International Application No. PCT/US2017/029866, 12 pages (Atty. Docket No. NRLA-009/01WO).	
	109.	International Search Report and Written Opinion mailed November 26, 2014 from International Application No. PCT/US2014/039239, 14 pages (Atty. Docket No. NRLA-003/01WO).	
	110.	International Search Report and Written Opinion mailed September 15, 2015 from International Application No. PCT/US2015/021492, 9 pages (Atty. Docket No. NRLA-004/01WO).	
	111.	Itti, L., and Koch, C. (2001). Computational modelling of visual attention. Nature Reviews Neuroscience, 2 (3), 194-203.	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT

(Use as many sheets as necessary)

Complete if Known	
Application Number	15/808,201
Filing Date	11-09-2017
First Named Inventor	Anatoli Gorchetchnikov
Art Unit	2611
Examiner Name	Not Yet Assigned
Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	112.	Itti, L., Koch, C., and Niebur, E. (1998). A Model of Saliency-Based Visual Attention for Rapid Scene Analysis, 1–6.	
	113.	Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., & LeCun, Y. (2009, September). What is the best multi-stage architecture for object recognition?. In Computer Vision, 2009 IEEE 12th International Conference on (pp. 2146-2153). IEEE.	
	114.	Kipfer, P., Segal, M., and Westermann, R. (2004). UberFlow: A GPU-Based Particle Engine. In Proceedings of the SIGGRAPH/Eurographics Workshop on Graphics Hardware 2004, pp. 115–122.	
	115.	Kolb, A., L. Latta, and C. Rezk-Salama. (2004). "Hardware-Based Simulation and Collision Detection for Large Particle Systems." In Proceedings of the SIGGRAPH/Eurographics Workshop on Graphics Hardware 2004, pp. 123–131.	
	116.	Kompella, Varun Raj, Luciw, Matthew, and Schmidhuber, J¨urgen. Incremental slow feature analysis: Adaptive low-complexity slow feature updating from high-dimensional input streams. Neural Computation, 24(11):2994–3024, 2012.	
	117.	Kowler, E. (2011). Eye movements: The past 25years. Vision Research, 51(13), 1457–1483. doi:10.1016/j.visres.2010.12.014	
	118.	Larochelle H., & Hinton G. (2012). Learning to combine foveal glimpses with a third-order Boltzmann machine. NIPS 2010, 1243-1251.	
	119.	LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010, May). Convolutional networks and applications in vision. In Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on (pp. 253-256). IEEE.	
	120.	Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755):788-791.	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT
(Use as many sheets as necessary)

Complete if Known	
Application Number	15/808,201
Filing Date	11-09-2017
First Named Inventor	Anatoli Gorchetchnikov
Art Unit	2611
Examiner Name	Not Yet Assigned
Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	121.	Lee, D. D., and Seung, H. S. (1997). "Unsupervised learning by convex and conic coding." Advances in Neural Information Processing Systems, 9.	
	122.	Legenstein, R., Wilbert, N., and Wiskott, L. Reinforcement learning on slow features of high-dimensional input streams. PLoS Computational Biology, 6(8), 2010. ISSN 1553-734X.	
	123.	Léveillé, J., Ames, H., Chandler, B., Gorchetchnikov, A., Mingolla, E., Patrick, S., and Versace, M. (2010) Learning in a distributed software architecture for large-scale neural modeling. BIONETICS10, Boston, MA, USA.	
	124.	Livitz G., Versace M., Gorchetchnikov A., Vasilkoski Z., Ames H., Chandler B., Leveille J. and Mingolla E. (2011) Scalable adaptive brain-like systems, The Neuromorphic Engineer, : 10.2417/1201101.003500 February 2011.	
	125.	Livitz, G., Versace, M., Gorchetchnikov, A., Vasilkoski, Z., Ames, H., Chandler, B., Léveillé, J., Mingolla, E., Snider, G., Amerson, R., Carter, D., Abdalla, H., and Qureshi, S. (2011) Visually-Guided Adaptive Robot (ViGuAR). Proceedings of the International Joint Conference on Neural Networks (IJCNN) 2011, San Jose, CA, USA.	
	126.	Lowe, D.G.(2004). Distinctive Image Features from Scale-Invariant Keypoints. Journal International Journal of Computer Vision archive Volume 60, 2, 91 – 110.	
	127.	Lu, Z.L., Liu, J., and Dosher, B.A. (2010) Modeling mechanisms of perceptual learning with augmented Hebbian re- weighting. Vision Research, 50(4). 375–390.	
	128.	Mahadevan, S. Proto-value functions: Developmental reinforcement learning. In Proceedings of the 22nd international conference on Machine learning, pp. 553–560. ACM, 2005.	
	129.	Meuth, J.R. and Wunsch, D.C. (2007) A SURVEY OF NEURAL COMPUTATION ON GRAPHICS PROCESSING HARDWARE. 22nd IEEE International Symposium on Intelligent Control, Part of IEEE Multi-conference on Systems and Control, Singapore, 1-3 October 2007, 5 pages.	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT

(Use as many sheets as necessary)

Complete if Known	
Application Number	15/808,201
Filing Date	11-09-2017
First Named Inventor	Anatoli Gorchetchnikov
Art Unit	2611
Examiner Name	Not Yet Assigned
Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	130.	Mishkin M, Ungerleider LG. (1982). "Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys," Behav Brain Res, 6 (1): 57–77.	
	131.	Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 25 February 2015.	
	132.	Moore, Andrew W and Atkeson, Christopher G. Prioritized sweeping: Reinforcement learning with less data and less time. Machine Learning, 13(1):103–130, 1993.	
	133.	Najemnik, J., and Geisler, W. (2009). Simple summation rule for optimal fixation selection in visual search. Vision Research. 49, 1286–1294.	
	134.	Notice of Allowance mailed 07/27/2016 from U.S. Application Serial No. 14/662,657 (Atty. Docket No. NRLA-005/01US).	
	135.	Notice of Allowance mailed 12/16/2016 from U.S. Application Serial No. 14/662,657 (Atty. Docket No. NRLA-005/01US).	
	136.	Oh, K.-S., and Jung, K. (2004). GPU implementation of neural networks. Pattern Recognition 37, pp. 1311–1314.	
	137.	Oja, E. (1982). Simplified neuron model as a principal component analyzer. Journal of Mathematical Biology 15(3), 267–273.	
	138.	Partial Supplementary European Search Report dated July 4, 2017 from European Application No. 14800348.6, 13 pages (Atty. Docket No. NRLA-003/01EP).	
	139.	Raijmakers, M.E.J., and Molenaar, P. (1997). Exact ART: A complete implementation of an ART network Neural networks 10 (4), 649-669.	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT

(Use as many sheets as necessary)

Complete if Known	
Application Number	15/808,201
Filing Date	11-09-2017
First Named Inventor	Anatoli Gorchetchnikov
Art Unit	2611
Examiner Name	Not Yet Assigned
Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	140.	Ranzato, M. A., Huang, F. J., Boureau, Y. L., & Lecun, Y. (2007, June). Unsupervised learning of invariant feature hierarchies with applications to object recognition. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on (pp. 1-8). IEEE.	
	141.	Raudies, F., Eldridge, S., Joshi, A., and Versace, M. (20 August 2014). Learning to navigate in a virtual world using optic flow and stereo disparity signals. Artificial Life and Robotics, DOI 10.1007/s10015-014-0153-1.	
	142.	Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. Nature Neuroscience, 2 (11), 1019-1025.	
	143.	Riesenhuber, M., & Poggio, T. (2000). Models of object recognition. Nature neuroscience, 3, 1199-1204.	
	144.	Rolfes, T. (2004). Artificial Neural Networks on Programmable Graphics Hardware. In Game Programming Gems 4, A. Kirmse, Ed. Charles River Media, Hingham, MA, pp. 373–378.	
	145.	Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In IEEE International Conference on Computer Vision (ICCV) 2011, 2564–2571.	
	146.	Ruesch, J. et al. (2008). Multimodal Saliency-Based Bottom-Up Attention A Framework for the Humanoid Robot iCub. 2008 IEEE International Conference on Robotics and Automation, pp. 962-965.	
	147.	Rumelhart D., Hinton G., and Williams, R. (1986). Learning internal representations by error propagation. In Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, MIT Press.	
	148.	Rumpf, M. and Strzodka, R. Graphics processor units: New prospects for parallel computing. In Are Magnus Bruaset and Aslak Tveito, editors, Numerical Solution of Partial Differential Equations on Parallel Computers, volume 51 of Lecture Notes in Computational Science and Engineering, pp. 89–134. Springer, 2005.	

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT (Use as many sheets as necessary)	Complete if Known	
	Application Number	15/808,201
	Filing Date	11-09-2017
	First Named Inventor	Anatoli Gorchetchnikov
	Art Unit	2611
	Examiner Name	Not Yet Assigned
	Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	149.	Salakhutdinov, R., & Hinton, G. E. (2009). Deep boltzmann machines. In International Conference on Artificial Intelligence and Statistics (pp. 448-455).	
	150.	Schaul, Tom, Quan, John, Antonoglou, Ioannis, and Silver, David. Prioritized experience replay. arXiv preprint arXiv: 1511.05952, 18 November 2015.	
	151.	Schmidhuber, J. (2010). Formal theory of creativity, fun, and intrinsic motivation (1990–2010). Autonomous Mental Development, IEEE Transactions on, 2(3), 230-247.	
	152.	Schmidhuber, Jürgen. Curious model-building control systems. In Neural Networks, 1991. 1991 IEEE International Joint Conference on, pp. 1458–1463. IEEE, 1991.	
	153.	Seibert, M., & Waxman, A.M. (1992). Adaptive 3-D Object Recognition from Multiple Views. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14 (2), 107-124.	
	154.	Sherbakov, L. and Versace, M. (2014) Computational principles for an autonomous active vision system. Ph.D., BOSTON UNIVERSITY, http://search.proquest.com/docview/1558856407	
	155.	Sherbakov, L. et al. (2012). CogEye: from active vision to context identification, youtube, retrieved from the Internet on 10-10-2017: URL://www.youtube.com/watch?v=i5PQk962B1k, 1 page.	
	156.	Sherbakov, L. et al. (2013). CogEye: system diagram module brain area function algorithm approx # neurons, retrieved from the Internet on 10-12-2017: URL://http://www-labsticc.univ-ubs.fr/~coussy/neucomp2013/index_fichiers/material/posters/NeuComp2013_final56x36.pdf, 1 page.	
	157.	Sherbakov, L., Livitz, G., Sohail, A., Gorchetchnikov, A., Mingolla, E., Ames, H., and Versace, M. (2013a) CogEye: An online active vision system that disambiguates and recognizes objects. NeuComp 2013.	

Examiner Signature:		**Date Considered**	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT (Use as many sheets as necessary)	Complete if Known	
	Application Number	15/808,201
	Filing Date	11-09-2017
	First Named Inventor	Anatoli Gorchetchnikov
	Art Unit	2611
	Examiner Name	Not Yet Assigned
	Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	158.	Sherbakov, L., Livitz, G., Sohail, A., Gorchetchnikov, A., Mingolla, E., Ames, H., and Versace, M (2013b) A computational model of the role of eye-movements in object disambiguation. Cosyne, Feb 28 - Mar 3, 2013. Salt Lake City, UT, USA.	
	159.	Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E.	
	160.	Snider, G., Amerson, R., Carter, D., Abdalla, H., Qureshi, S., Laveille, J., Versace, M., Ames, H., Patrick, S., Chandler, B., Gorchetchnikov, A., and Mingolla, E. (2010) Adaptive Computation with Memristive Memory. IEEE Computer, Volume 44(2).	
	161.	Spratling, M. W. (2008). Predictive coding as a model of biased competition in visual attention. Vision Research, 48(12):1391-1408.	
	162.	Spratling, M. W. (2012). Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. Neural Computation, 24(1):60-103.	
	163.	Spratling, M. W., De Meyer, K., and Kompass, R. (2009). Unsupervised learning of overlapping image components using divisive input modulation. Computational intelligence and neuroscience.	
	164.	Sprekeler, H. On the relation of slow feature analysis and laplacian eigenmaps. Neural Computation, pp. 1–16, 2011.	
	165.	Sutton, Richard S and Barto, Andrew G. Reinforcement learning: An introduction. MIT press, 1998.	
	166.	TEDx Fulbright, Invited talk, Washington DC, 4/5/2014.	

Examiner Signature:		**Date Considered**	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT (Use as many sheets as necessary)	Complete if Known	
	Application Number	15/808,201
	Filing Date	11-09-2017
	First Named Inventor	Anatoli Gorchetchnikov
	Art Unit	2611
	Examiner Name	Not Yet Assigned
	Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	167.	Tong, F., Ze-Nian Li, (1995). Reciprocal-wedge transform for space-variant sensing," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.17, no.5, pp.500 - 51. doi: 10.1109/34.391393	
	168.	Torralba, A., Oliva, A., Castelhano, M.S., Henderson, J.M. (2006). Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. Psychological Review, 113(4).766–786.	
	169.	Van Hasselt, Hado, Guez, Arthur, and Silver, David. Deep reinforcement learning with double q-learning. arXiv preprint arXiv: 1509.06461, 22 September 2015.	
	170.	Versace, M. (2006) From spikes to interareal synchrony: how attentive matching and resonance control learning and information processing by laminar thalamocortical circuits. NSF Science of Learning Centers PI Meeting, Washington, DC, USA.	
	171.	Versace, M., (2010) Open-source software for computational neuroscience: Bridging the gap between models and behavior. In Horizons in Computer Science Research,Volume 3.	
	172.	Versace, M., Ames, H., Léveillé, J., Fortenberry, B., and Gorchetchnikov, A. (2008) KInNeSS: A modular framework for computational neuroscience. Neuroinformatics, 2008 Winter; 6(4):291-309. Epub 2008 Aug 10.	
	173.	Versace, M., and Chandler, B. (2010) MoNETA: A Mind Made from Memristors. IEEE Spectrum, December 2010.	
	174.	Webster, Bachevalier, Ungerleider (1994). Connections of IT areas TEO and TE with parietal and frontal cortex in macaque monkeys. Cerebal Cortex, 4(5), 470-483.	
	175.	Wiskott, Laurenz and Sejnowski, Terrence. Slow feature analysis: Unsupervised learning of invariances. Neural Computation, 14(4):715–770, 2002.	

156249486 v1

Examiner Signature:		**Date Considered**	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

Electronic Acknowledgement Receipt

EFS ID:	31168347
Application Number:	15808201
International Application Number:	
Confirmation Number:	1077
Title of Invention:	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD
First Named Inventor/Applicant Name:	Anatoli Gorchetchnikov
Customer Number:	58249
Filer:	Nisha Somnath/Donna Doyle
Filer Authorized By:	Nisha Somnath
Attorney Docket Number:	NRLA-001/03US 320730-2040
Receipt Date:	08-DEC-2017
Filing Date:	
Time Stamp:	19:55:08
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	NRLA-001_03US-Transmittal.pdf	24916 c5435c41434c7b763dc9bd93c6b397be664a71b1	no	2

Warnings:

Information:					
2	Foreign Reference	EP1224622B1.pdf	2058261 daae9c30d6cf1ae32ffd606fbdf435a50d03c1eb	no	33
Warnings:					
Information:					
3	Foreign Reference	WO14190208.pdf	6849847 2a6335da5df4ddd9826283f43347a07b365c62b8	no	76
Warnings:					
Information:					
4	Foreign Reference	WO14204615.pdf	3157209 e188a783ad5cb9a1f45a59badb8def893f26ea37	no	39
Warnings:					
Information:					
5	Foreign Reference	WO15143173.pdf	8162745 e164d07a4472e31977e81ff139475e1fc938e4ed	no	90
Warnings:					
Information:					
6	Foreign Reference	WO16014137.pdf	5133789 bea3c8d923ef37d36e0d538d9178ede1f29941c4	no	69
Warnings:					
Information:					
7	Non Patent Literature	Adelson_PyramidMethodsInImageProcessing.pdf	1808321 e18f001b7a96c8968f5502e7388386c134e1f500	no	9
Warnings:					
Information:					
8	Non Patent Literature	Aggarwal_OnTheSurprisingBehaviorOfDistanceMetricsInHighDimensionalSpace.pdf	203505 4215a61b1f53997337011c7325fae467529223ba	no	15
Warnings:					
Information:					

9	Non Patent Literature	Ames_PersuadingComputersToActMoreLikeBrains.pdf	882347 34df80ee8de3d8732449b2285cc0cee26dc53367	no	25
Warnings:					
Information:					
10	Non Patent Literature	Ames_TheAnimat-NewFrontiersInWholeBrainModeling.pdf	1011465 e4094e6e5933fc29ac66c13014f30e0f4322bb08	no	4
Warnings:					
Information:					
11	Non Patent Literature	ArtificialIntelligenceAsAService.pdf	2159685 6cdb6c02131e9f2c17e43236e01934d015ae1f21	no	22
Warnings:					
Information:					
12	Non Patent Literature	Aryananda.pdf	1296842 fe12f3c9a67c8117e0a91b7949790ccf13195641	no	6
Warnings:					
Information:					
13	Non Patent Literature	Baraldi_SimplifiedART-ANewClassOfARTAlgorithms.pdf	2057373 e0ca2b21919216738b319f4e33ea729e91c1b8ba	no	42
Warnings:					
Information:					
14	Non Patent Literature	Baraldi_ConstructiveFeedforwardARTClusteringNetworks-PartI.pdf	545879 8adebb7a39aa5ff2cd2867e49f5115185c3db9b7	no	17
Warnings:					
Information:					
15	Non Patent Literature	Baraldi_ConstructiveFeedforwardARTClusteringNetworks-PartII.pdf	692505 fb65f719779f0f177d3cc8c67af75e60689166c9	no	16
Warnings:					
Information:					

16	Non Patent Literature	Baraldi_FuzzyCombinationOfKohonensAndARTNeuralNetworkModelsToDetectStatistical.pdf	954372 832221d0202f42507e15bf28f45035eb37459764	no	6
Warnings:					
Information:					
17	Non Patent Literature	Bengio_RepresentationLearning-AReviewAndNewPerspectives.pdf	852491 e7c6ede71f16962894579cc58c16d71dd5a6df4	no	30
Warnings:					
Information:					
18	Non Patent Literature	Berenson.pdf	2130174 ad50ec895ee8c56227dfabcd9e388a7377a5bf5c	no	9
Warnings:					
Information:					
19	Non Patent Literature	Bernhard.pdf	494900 8aa1cbe53f4ab50a9d02a3ca4eeb6023903418eb	no	8
Warnings:					
Information:					
20	Non Patent Literature	Besl_Three-dimensionalObjectRecognition.pdf	15778862 26ff88558940877eb3a3d9f3d4a0a947815ce066	no	71
Warnings:					
Information:					
21	Non Patent Literature	Bohn.pdf	184181 af86f618c707ea03aec8f235f77de0e75d0ede42	no	4
Warnings:					
Information:					
22	Non Patent Literature	Bradski_FastLearning.pdf	3017194 a662118861537b4fa03a38807777580919a49050	no	28
Warnings:					
Information:					

23	Non Patent Literature	BrainInspiredComputing_Boston2010.pdf	21467 f8d229582ce1aca4440876f9b57c19a693e3e2e7	no	1
Warnings:					
Information:					
24	Non Patent Literature	Canny.pdf	16505538 a2d00698cf121c8eed088f827da8653da1bbb0d0	no	20
Warnings:					
Information:					
25	Non Patent Literature	CarpenterAMassivelyParallel.pdf	5317488 a2ef3bfcf8317dc3e1d1ff99b40e7dbe1910c2bf	no	62
Warnings:					
Information:					
26	Non Patent Literature	Carpenter_AdaptiveResonanceTheory.pdf	175098 192ff1545d40bca681f0031679fc6ba44ebecc94	no	12
Warnings:					
Information:					
27	Non Patent Literature	Carpenter_FuzzyART.pdf	20929851 696241ee2837894bf0c3db088dc7fe6cb4c62fa4	no	13
Warnings:					
Information:					
28	Non Patent Literature	Carpenter_TheArtOfAdaptivePatternRecognitionByASelf-organizingNeuralNetwork.pdf	8037230 ed8100c0cf03bddeb15466622503359000 2686c9	no	27
Warnings:					
Information:					
29	Non Patent Literature	Coifman_DiffusionWavelets.pdf	2127458 834cf7ea9797860bdc9a9849b1728566705d8431	no	66
Warnings:					
Information:					

30	Non Patent Literature	Coifman_GeometricDiffusionsAsAToolForHarmonicAnalysisAndStructureDefinitionOfData-DiffusionMaps.pdf	645237 9867ef6d784dd52a7861e41692b4fb2f558a8e36	no	21
Warnings:					
Information:					
31	Non Patent Literature	Davis.pdf	968444 4b7147cddfb7415fca9fcd39cdf788f73120b231	no	121
Warnings:					
Information:					
32	Non Patent Literature	Dosher_MechanismsOfPerceptualAttentionInPrecuingOfLocation.pdf	1228600 acd1c5235ac1644f8c4da39505514232d91e4fd8	no	24
Warnings:					
Information:					
33	Non Patent Literature	Ellias_1.pdf	4925050 154e6b270dd51fe335dfc0327bf2c3f4479aacf0	no	31
Warnings:					
Information:					
34	Non Patent Literature	NRLA00201EP_EESR_2017-06-01.pdf	9489584 507369974548c517836881a927ee5cfb5dfbcb35	no	10
Warnings:					
Information:					
35	Non Patent Literature	NRLA00301EP_SearchReport_2017-10-12.pdf	1029213 374ac0ceffec5219ddc9681f2188519e90575b18	no	12
Warnings:					
Information:					
36	Non Patent Literature	NRLA00401EP_SearchReport_2017-10-23.pdf	974846 cdba4a9209b3a739e0ea5daee9b07d08ff5fc9dd	no	8
Warnings:					
Information:					

37	Non Patent Literature	Fazl_View-invariantObjectCategoryLearningRecognitionAndSearch.pdf	1409136 4a7a2c705f6a1c11b8599d250c7ee63d969accc3	no	68
Warnings:					
Information:					
38	Non Patent Literature	Foldiak_FormingSparseRepresentationsByLocalAnti-HebbianLearning.pdf	10367340 6e296fd698d2b5f72baa6255fada70d17af3f0e2	no	6
Warnings:					
Information:					
39	Non Patent Literature	Friston_PerceptionsAsHypothesesSaccadesAsExperiments.pdf	3199877 d0e8396ad76e4618218d8360369922917b0c3acf	no	20
Warnings:					
Information:					
40	Non Patent Literature	Galbraith_ANeuralNetwork-basedExploratoryEarningAndMotorPlanningSystemForCo-robots.pdf	1615298 87dd743051234961c7864a2cba452c1c98091154	no	10
Warnings:					
Information:					
41	Non Patent Literature	George_TowardsAMathematicalTheoryOfCorticalMicro-circuits-PLoSComputationalBiology.pdf	2551942 c196a91d7c24588c04d8a8788b20e88d622e0d19	no	26
Warnings:					
Information:					
42	Non Patent Literature	Georgii.pdf	526830 62bd552e0eefd8e330e7b21b4e45830eb94c8c43	no	10
Warnings:					
Information:					
43	Non Patent Literature	gorchetchnikov-hasselmo-05.pdf	852416 e6519c303dda858d65db5d97e546642d3d840d81	no	22
Warnings:					
Information:					

44	Non Patent Literature	GorchASimpleRulepdf.pdf	4073667 3713f55ba31cb8ed2945c48cff609a16adff6e23	no	6
Warnings:					
Information:					
45	Non Patent Literature	GVH_modelstdp.pdf	308252 90f4ef44e84be07c1fabd8c6bc6060fb96af79d2	no	9
Warnings:					
Information:					
46	Non Patent Literature	GVH_spatially.pdf	1114226 e97dba30d4716e69ba512bda556c8a921282f4ea	no	6
Warnings:					
Information:					
47	Non Patent Literature	Grossberg_ContourEnhancementShort-termMemoryAndConstanciesInReverberatingNeuralNetworks.pdf	4886742 2a7eac57672728a4ffaad6177faed6044b44fd77	no	46
Warnings:					
Information:					
48	Non Patent Literature	Grossberg_ANeuralSystemForNaturalSceneClassification.pdf	2616544 dc5aa5f1f8ebbe5352038e2cfd241c66e8d1353e	no	19
Warnings:					
Information:					
49	Non Patent Literature	Grossberg_SpikesSynchronyAndAttentiveLearningByLaminarThalamocorticalCircuits.pdf	2402871 6a35a1195765f58237769471f75f39f2d1347a7f	no	52
Warnings:					
Information:					
50	Non Patent Literature	Hagen.pdf	306417 b917912ca3fad496bf4a525e856f2ad44557a1a2	no	12
Warnings:					
Information:					

51	Non Patent Literature	Hasselt_Double_qlearning.pdf	167325 610b6ae8774de91f6430748e0bce0a08ca18133f	no	9
Warnings:					
Information:					
52	Non Patent Literature	Hinton_AFastLearningAlgorithmForDeepBeliefNets.pdf	830958 75d6e2f563af4812f9a03199e4cb8fed84b3400c	no	28
Warnings:					
Information:					
53	Non Patent Literature	hodgkin5.pdf	13074543 b27b2284c2ba779026d1439bcd0f1a568712bba5	no	45
Warnings:					
Information:					
54	Non Patent Literature	hopfield82.pdf	1326555 18c3fb0171565436d228eb3cc25d37d4a0ee9494	no	6
Warnings:					
Information:					
55	Non Patent Literature	Ilie_1.pdf	305021 b888573992c9b85b5f0d3417eadde4978dc73f2f	no	9
Warnings:					
Information:					
56	Non Patent Literature	PCTUS2014039162InternationalPreliminaryReportOnPatentability.pdf	6921125 b913df4d542ff3f8c8b86d0714fc540b2a95831b	no	7
Warnings:					
Information:					
57	Non Patent Literature	PCTUS2014039239InternationalPreliminaryReportOnPatentability.pdf	10252126 e1f4343d209a34277263a13cb098547ca6e7e03b	no	8
Warnings:					
Information:					

58	Non Patent Literature	NRLA00601WO_IPRP_2016-14-08.pdf	693848 d1dcfc60a20c09888055f4441a916f64a40a075b	no	7
Warnings:					
Information:					
59	Non Patent Literature	NRLA00201WO_ISRandWO_2015-02-18.pdf	2000248 b6c90138832f95a7ff87fba93df8124ab4eadc99	no	12
Warnings:					
Information:					
60	Information Disclosure Statement (IDS) Form (SB08)	NRLA-001_03US-SB08.pdf	358294 9e187ec4fe5ce30e680d4934ef00d31e34bb721b	no	16
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
			Total Files Size (in bytes):	203993568	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Attorney Docket No.: NRLA-001/03US 320730-2040 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	Anatoli Gorchetchnikov et al.	Confirmation No.:	1077
Application No.:	15/808,201	Group Art Unit:	2611
Filed:	November 9, 2017	Examiner:	Not Yet Assigned

FOR: **GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD**

VIA EFS or
Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. §1.97(b)

In accordance with the duty of disclosure set forth in 37 C.F.R. §1.56, Applicant(s) hereby submits the following information in conformance with 37 C.F.R. §§1.97 and 1.98.

☒ Pursuant to 37 C.F.R. §1.98, a copy of each non-US patent document cited in the attached Form PTO/SB/08 is enclosed.

☒ No copies of any U.S. Office Actions cited on the accompanying Form PTO/SB/08 are being provided because they are available on PAIR. Copies will be provided to the Examiner upon request.

☐ No copies of the publications listed on the attached Form PTO/SB/08 are being provided pursuant to 37 C.F.R. §1.98(d) because the publications were previously cited by or submitted to the Office in prior Application Serial No. _____ to which the above-identified application claims priority under 35 U.S.C. §120.

☒ No copies of any U.S. patents or U.S. patent application publications listed on the attached Form PTO/SB/08 are being provided pursuant to 37 C.F.R. §1.98.

☐ Publication(s) _____ listed on the attached Form PTO/SB/08 were cited in a foreign search or examination report corresponding to _____ application serial no. _____ and mailed on _____.

☐ Enclosed is a copy of a non-English publication(s) _____ Pursuant to §609 of the M.P.E.P., Applicant submits the attached foreign search or examination report, which cites such non-English language publication(s).

☐ Enclosed is a copy of a non-English publication(s) _____ English language publication _____ (copy enclosed) claims priority from this non-English publication.

☐ Enclosed is an explanation of non-English publication(s) _____ for which an English translation is not available.

☐ Enclosed is an English translation of non-English publication(s) _____ cited in the attached Form PTO/SB/08.

☐ Enclosed is a copy of pending patent Application Serial No. _____.

This Information Disclosure Statement is filed within any one of the following time periods:

☐ within three months from the filing date of this national application other than a CPA under 37 C.F.R. § 1.53(d);

☐ within three months from the date of entry of the national stage as set forth in 37 C.F.R. §1.491 in this international application;

☒ before the mailing date of a first office action on the merits; or

☐ before the mailing of a first office action after the filing of a request for continued examination under 37 C.F.R. § 1.114.

It is respectfully requested that the Examiner consider the above-noted information and return an initialed copy of the attached Form PTO/SB/08 to the undersigned.

Dated: 12/8/2017

Respectfully submitted,
COOLEY LLP

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW, Suite 700
Washington, DC 20004

Tel: (617) 937-2412
Fax: (202) 842-7899

By: /Nisha Somnath/
Nisha Somnath
Reg. No. L1194

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 224 622 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
10.11.2004 Bulletin 2004/46

(21) Application number: **00968390.5**

(22) Date of filing: **22.09.2000**

(51) Int Cl.[7]: **G06T 15/70**

(86) International application number:
PCT/US2000/026081

(87) International publication number:
WO 2001/022370 (29.03.2001 Gazette 2001/13)

(54) **METHOD AND APPARATUS FOR RAPID VISUALIZATION OF THREE-DIMENSIONAL SCENES**

VERFAHREN UND VORRICHTUNG ZUR SCHNELLEN VISUALISIERUNG VON DREIDIMENSIONALEN SZENEN

PROCEDE ET APPAREIL DE TRAITEMENT RAPIDE DE PROGRAMMES A BASE DE SCENES

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **24.09.1999 US 156054 P**

(43) Date of publication of application:
24.07.2002 Bulletin 2002/30

(73) Proprietor: **Sun Microsystems, Inc.**
Santa Clara, California 95054 (US)

(72) Inventors:
- **DEERING, Michael, F.**
  Los Altos, CA 94024 (US)
- **SOWIZRAL, Henry**
  Los Altos, CA 94022-1240 (US)
- **RUSHFORTH, Kevin**
  San Jose, CA 95124-2207 (US)
- **TWILLEAGER, Doug**
  Campbell, CA 95008-5112 (US)

(74) Representative: **Davies, Simon Robert**
D Young & Co
120 Holborn
London, EC1N 2DY (GB)

(56) References cited:
US-A- 5 657 479 **US-A- 5 896 139**

- **RIKK CAREY ET AL: "The virtual reality modeling language" INTERNATIONAL STANDARD ISO/IEC 14772-1:1997, [Online] 1997, XP002133320 Retrieved from the Internet: <URL:http://www.vrml.org/Specifications/VRML97> [retrieved on 2001-02-28]**
- **ARIKAWA M ET AL: "DYNAMIC LOD FOR QOS MANAGEMENT IN THE NEXT GENERATION VRML" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON MULTIMEDIA COMPUTING AND SYSTEMS,US,LOS ALAMITOS, CA, 17 June 1996 (1996-06-17), pages 24-27, XP002060163**
- **"DEALER INSTRUCTION PROCESSING UNIT GOVERNOR - USE OF THREAD STRUCTURE AND DIRECTED ACYCLIC GRAPH TO CREATE DLIST/SLIST" IBM TECHNICAL DISCLOSURE BULLETIN,US,IBM CORP. NEW YORK, vol. 36, no. 6A, 1 June 1993 (1993-06-01), pages 579-581, XP000373101 ISSN: 0018-8689**

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 1 224 622 B1

Printed by Jouve, 75001 PARIS (FR)

Description

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0001] This invention relates generally to the field of computer graphics and, more particularly, to graphics systems and software that manage and render three-dimensional graphics data.

2. Description of the Related Art

[0002] A computer system typically relies upon its graphics system for producing visual output on the computer screen or display device. Early graphics systems were only responsible for taking what the processor produced as output and displaying it on the screen. In essence, they acted as simple translators or interfaces. Modern graphics systems, however, incorporate graphics processors with a great deal of processing power. They now act more like coprocessors rather than simple translators. This change is due to the recent increase in both the complexity and amount of data being sent to the display device. For example, modern computer displays have many more pixels, greater color depth, and are able to display more complex images with higher refresh rates than earlier models. Similarly, the images displayed are now more complex and may involve advanced techniques such as anti-aliasing and texture mapping.

[0003] As a result, without considerable processing power in the graphics system, the CPU would spend a great deal of time performing graphics calculations. This could rob the computer system of the processing power needed for performing other tasks associated with program execution and thereby dramatically reduce overall system performance. With a powerful graphics system, however, when the CPU is instructed to draw a box on the screen, the CPU is freed from having to compute the position and color of each pixel. Instead, the CPU may send a request to the video card stating "draw a box at these coordinates." The graphics system then draws the box, freeing the processor to perform other tasks.

[0004] Generally, a graphics system in a computer (also referred to as a graphics system) is a type of video adapter that contains its own processor to boost performance levels. These processors are specialized for computing graphical transformations, so they tend to achieve better results than the general-purpose CPU used by the computer system. In addition, they free up the computer's CPU to execute other commands while the graphics system is handling graphics computations. The popularity of graphical applications, and especially multimedia applications, has made high performance graphics systems a common feature of computer systems. Most computer manufacturers now bundle a high performance graphics system with their systems.

[0005] Since graphics systems typically perform only a limited set of functions, they may be customized and therefore far more efficient at graphics operations than the computer's general-purpose central processor. While early graphics systems were limited to performing two-dimensional (2D) graphics, their functionality has increased to support three-dimensional (3D) wire-frame graphics, 3D solids, and now includes support for three-dimensional (3D) graphics with textures and special effects such as advanced shading, fogging, alpha-blending, and specular highlighting.

[0006] To take advantage of the new capabilities of both graphics systems and modem CPUs in general, graphics application program interfaces ("APIs") have been developed. An API is a set of routines, protocols, and/or tools for building software applications. An API attempts to simplify the task of developing a program by providing building blocks needed by programmers to create their application. One example of a popular API is Microsoft Corporation's Win32 API.

[0007] While graphics API have been successful in allowing programmers to rapidly develop graphical applications, the recent increase in the complexity of the scenes being rendered is placing ever greater demands on the CPUs and graphics systems that are executing the applications. Traditionally, when tasks become too computationally intensive for a single CPU, multiple CPU systems are used. These multiple CPU systems utilize parallel processing to divide the computations among two or more processors. However, typical graphics APIs are not well suited to allow parallel processing, particularly over a network. Thus, a system and method for efficiently managing the creation, updating, and rendering of a complex scene is needed. In particular, a system and method capable of being implemented in an API and capable of supporting distributed processing in a networked setting is also desired.

[0008] US 5896139 discloses a system, method and computer program product for a scene graph for optimizing rendering performance. The nodes of an original scene graph are rearranged to minimize a number of state changes while rendering the scene graph. The shapes of the original scene graph are further broken down into a plurality of triangular strip sets, which leads to a better rendering performance.

NVIDIA Ex1002 Page 160 of 1081

SUMMARY OF THE INVENTION

[0009] The problems identified above may at least in part be solved by a system and method for rapid processing of scene graph-based data as set out in the appended claims. In one embodiment, the system generates a parallel structure for the scene graph-based data. The parallel structure may include both objects and threads. Advantageously, the system may utilize a parallel structure for rendering and thereby avoid repeated traversals of the scene graph in its hierarchy (i.e., tree) form. This parallel structure may be inplemented in an API such that it is effectively unseen by graphics application programmers, who continues to use the scene graph structure to create graphics applications.

[0010] In one embodiment, the system may be configured to utilize a scens graph directly. In another embodiment, the system may be configured to utilize a scene graph as a source for ancillary structures. The ancillary structures may aid the system in rendering and executing the scene-graph-based program. The system may also include support for messaging between threads and objects, messaging with time and/or event stamps, epochs to ensure consistency, and ancillary structures such as render-bins, geometry structures, and rendering environment structures.

[0011] A computer program is also contemplated. In one embodiment, the computer program comprises a plurality of instructions configured to receive a scene graph and traverse the scene graph to generate a plurality of structures and threads corresponding to the scene graph. The scene graph may include information representing a plurality of three-dimensional objects. The scene graph may also include a number of different types of data, including, behavior data for the three-dimensional objects, sound data, hauptics (e.g., force-feed back data) appearance data, geometry data, environmental data (e.g., lighting, fog), and other data. The information may include appearance data, geometry data, and environmental data. Each structure may be an object that manages selected data from the scene graph, and the plurality of threads may be exacutable to render one or more frames corresponding to the scene graph. In some embodiments the threads may be configured to generate messages to specify state change, and the messages can be multicast to multiple structures or unicast to a single structure. Each structure may have a corresponding update thread configured to update the structure.

One of the parallel structures created is preferably a render bin that is configured to receive and store references to particular geometry data that is to be rendered in the render bin. The render bin may have one or more render associated with it, thereby enabling parallel rendering utilizing multiplier processors.

[0012] Advantageously, each structure may be configured to manage (and optionally optimize) its own data. Thus one structure may manage all transform data, and may optimize multiple transforms by collapsing them together.

[0013] As noted above, a method for managing and rendering a scene graph is also contemplated. In one embodiment, the method may include generating a scene graph, wherein the scene graph comprises information representing a plurality of three-dimensional objects; and traversing the scene graph to generate a parallel set of structures and threads corresponding to the scene graphs, wherein each structure comprises an object that manages selected data from the scene graph, and wherein the plurality of threads are executable to render one or more frames corresponding to the scene graph. The method may be implemented in software, hardware, or a combination thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The foregoing, as well as other objects, features, and advantages of this invention may be more completely understood by reference to the following detailed description when read together with the accompanying drawings in which:

Figure 1 illustrates one embodiment of a computer system that may be used to implement the system and method described herein;
Figure 2 is a simplified block diagram of the computer system of Figure 1;
Figure 3 is a diagram illustrated one embodiment of a scene graph;
Figure 4 is a diagram illustrated one embodiment of a scene graph with parallel structures for managing and rendering;
Figure 5 is a flowchart of one embodiment of a method for generating the parallel structure of Figure 4;
Figure 6 is a diagram of another embodiment of a scene graph;
Figure 7 is a diagram of one embodiment of structure optimization;
Figure 8 is a diagram of one embodiment of a list of current work threads;
Figure 9 is a diagram illustrating a number of example configurations for screen and canvas lists.

[0015] While the invention is susceptible to various modifications and alternative forms, specific embodiments thereof are shown by way of example in the drawings and will herein be described in detail. It should be understood, however, that the drawings and detailed description thereto are not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and

NVIDIA Ex1002 Page 161 of 1081

scope of the present invention as defined by the appended claims. The word "may" is used in this application in a permissive sense (i.e., having the potential to, being able to), not a mandatory sense (i.e., must). Similarly, the word include, and derivations thereof, are used herein to mean "including, but not limited to."

DETAILED DESCRIPTION OF SEVERAL EMBODIMENTS

Computer System - Figure 1

[0016] Referring now to Figure 1, one embodiment of a computer system 10 that may used to implement the system and method described above is illustrated. The computer system may be used as the basis for any number of various systems, including a traditional desktop personal computer, a laptop computer, a network PC, an Internet appliance, a television, including HDTV systems and interactive television systems, personal digital assistants (PDAs), and other device which displays 2D and or 3D graphics.

[0017] As shown in the figure, computer system 10 comprises a system unit 12 and a video monitor or display device 14 coupled to the system unit 12. The display device 14 may be any of various types of display monitors or devices (e.g., a CRT, LCD, or gas-plasma display). Various input devices may be connected to the computer system, including a keyboard 16 and/or a mouse 18, or other input device (e.g., a trackball, digitizer, tablet, six-degree of freedom input device, head tracker, eye tracker, data glove, body sensors, etc.). Application software may be executed by the computer system 10 to display 3D graphical objects on display device 14. The application software may be stored in memory, read from a server using a network connection, or read from a storage medium such as a computer diskette, CD-ROM, DVD-ROM, or computer tape.

Computer System Block Diagram - Figure 2

[0018] Referring now to Figure 2, a simplified block diagram illustrating one embodiment of the computer system of Figure 1 is shown. Elements of the computer system that are not necessary for an understanding of the present invention are not shown for convenience. As shown, the computer system 10 includes a central processing unit (CPU) 22 coupled to a high-speed memory bus or system bus 24. A system memory 26 may also be coupled to high-speed bus 24.

[0019] Host processor 22 may comprise one or more processors of varying types, e.g., microprocessors, multiprocessors and CPUs. The system memory 26 may comprise any combination of different types of memory subsystems, including random access memories, (e.g., static random access memories or "SRAMs", synchronous dynamic random access memories or "SDRAMs", and Rambus dynamic access memories or "RDRAM", among others) and mass storage devices. The system bus or host bus 24 may comprise one or more communication or host computer buses (for communication between host processors, CPUs, and memory subsystems) as well as specialized subsystem buses.

[0020] A 3D graphics system or graphics system 28 according to the present invention is coupled to the high-speed memory bus 24. The 3D graphics system 28 may be coupled to bus 24 by, for example, a crossbar switch or other bus connectivity logic. It is assumed that various other peripheral devices, or other buses, may be connected to the high-speed memory bus 24. It is noted that the 3D graphics system may be coupled to one or more of the buses in computer system 10 and/or may be coupled to various types of buses. In addition, the 3D graphics system may be coupled to a communication port and thereby directly receive graphics data from an external source, e.g., the Internet or a network. As shown in the figure, display device 14 is connected to the 3D graphics system 28 comprised in the computer system 10.

[0021] Host CPU 22 may transfer information to and from the graphics system 28 according to a programmed input/output (I/O) protocol over host bus 24. Alternately, graphics system 28 may access the memory subsystem 26 according to a direct memory access (DMA) protocol or through intelligent bus mastering.

[0022] A graphics application program conforming to an application programming interface (API) such as OpenGL or Java 3D may execute on host CPU 22 and generate commands and data that define a geometric primitive (graphics data) such as a polygon for output on display device 14. As defined by the particular graphics interface used, these primitives may have separate color properties for the front and back surfaces. Host processor 22 may transfer these graphics data to memory subsystem 26. Thereafter, the host processor 22 may operate to transfer the graphics data to the graphics system 28 over the host bus 24. In another embodiment, the graphics system 28 may read in geometry data arrays over the host bus 24 using DMA access cycles. In yet another embodiment, the graphics system 28 may be coupled to the system memory 26 through a direct port, such as the Advanced Graphics Port (AGP) promulgated by Intel Corporation.

[0023] The graphics system may receive graphics data from any of various sources, including the host CPU 22 and/or the system memory 26, other memory, or from an external source such as a network, e.g., the Internet, or from a broadcast medium, e.g., television, or from other sources.

NVIDIA Ex1002 Page 162 of 1081

[0024] One embodiment of a graphics system may comprise a graphics processor coupled to a display buffer. As used herein, a display buffer is a buffer that stores information (e.g., pixels or samples that may be used to form pixels) that represents a particular frame to be output and displayed on a display device. For example, a traditional frame buffer may be considered to be a display buffer. The contents of the display buffer are typically output to digital to analog converters (DACs) that convert the digital information into analog video signals for display on a display device. Depending upon the configuration, the graphics processor (or processors, in the case of a parallel processing system) is configured to execute instructions and/or process data received from main system memory and/or the host CPU.

[0025] Computer system 10 may also include a network interface (not shown) to allow the computer system to send and receive data from a computer network. In one embodiment, the computer system may be configured to distribute tasks associated with the management and rendering of a three-dimensional scene graph to other computers on the network. The tasks may be distributed by sending objects (e.g., data and threads) to other computers on the network (i.e., each utilizing its own non-shared memory). The computers may communicate with each other via a system of messages (as explained in greater detail below).

Scene Graph - Figure 3

[0026] As noted above, scene graphs are hierarchies of objects and graphics data that define a particular three-dimensional scene or world. For example, the top node may represent an entire building and all the furnishings and people that occupy the building. As the scene graph is traversed, the next level of group nodes may represent particular rooms within the building. Traversing the scene graph one more level may yield group nodes that each represent one person or furniture object (e.g., a chair) in a particular room. Each group node may also have transform node that translates and/or rotates any nodes below the transform node (i.e., child nodes). While many levels of the scene graph are possible, eventually a traversal of the scene graph will terminate at a leaf node. Leaf nodes typically represents a certain object (e.g., a tea cup or a chair) or a portion of a larger object (e.g., a person's hand). The leaf node typically has one or more pointers to graphics data the described the object. For example, in embodiment the leaf node may have pointers to a data file that includes polygons (or NURBS - Non-Uniform Rational B-Splines) defining the shape of the object and texture maps that define the appearance of the object. The leaf node may also have pointers to a bounding box.

[0027] Before the scene graph is submitted for rendering, it is advantageous to perform "object culling". Object culling refers to the process of selectively removing objects from the scene graph. In most implementations of object culling, objects are removed either because they are outside the current view frustum, or because they are occluded (either partially or fully). Object is particularly useful for complex scene graphs that have large number of objects. Since current graphics hardware may be unable to achieve satisfactory frame rates when rendering all of the objects, object culling attempts to remove the objects that cannot be seen from the current viewpoint/orientation. Thus object culling attempts to reduce wasting graphics pipeline bandwidth by rendering only those objects that are actually visible to the viewer. For example, assuming that a particular scene graph represents a virtual model of an entire city, if the current viewpoint is close to the base of a particular building, that particular building will obscure those behind it. Thus, assuming that the building is opaque and that no shadows or reflections of other structures behind the particular building are visible, then there is no reason to devote rendering resources to rendering those obscured structures.

[0028] Turning now to Figure 3, one embodiment of a scene graph 50 is shown. In this example, scene graph 50 is created from instances of Java 3D classes, but other types of scene graphs are also possible and contemplated (e.g., VRML scene graphs). Scene graph 50 is assembled from objects that define the geometry, sound, lights, location, orientation, and appearance of visual and audio objects.

[0029] As illustrated in the figure, scene graph 50 may be thought of as a data structure having nodes and arcs. A node represents a data element, and an arc represents the relationship between data elements. In the figure, nodes 54-66 and 74 of scene graph 50 are the instances of Java 3D classes. The arcs represent two types of relationships between the Java 3D class instances. The first type of relationship is a parent-child relationship. For example, in this embodiment a group node such as branch group node 60 can have any number of children (such as Shape3d leaf node 64), but only one parent (i.e., local node 54). Similarly, a leaf node has one parent but no children. These relationships are indicated in the figure by solid lines. The second type of relationship is a reference. A reference associates a particular node component object with a scene graph node. Node component objects define the geometry and appearance attributes used to render the visual objects. References are indicated in the figure by dashed lines.

[0030] As shown in the figure, Java 3D™ scene graph 50 is constructed of node objects in parent-child relationships forming a tree structure. In a tree structure, one node (e.g., VirtualUniverse node 52 in the figure) is the root. Other nodes are accessible by following arcs from the root. Scene graph 50 is formed from the trees rooted at locale object 54. The node components and reference arcs are not part of the scene graph tree. Since only one path exists from the root of a tree to each of the leaves, there is only one path from the root of a scene graph to each leaf node. The path from the root of scene graph 50 to a specific leaf node is called the leaf node's "scene graph path." Since a scene

NVIDIA Ex1002 Page 163 of 1081

graph path leads to exactly one leaf, there is one scene graph path for each leaf in the scene graph 50.

[0031] Each scene graph path in a Java 3D scene graph completely specifies the state information of its leaf. State information includes the location, orientation, and size of a visual object. Consequently, the visual attributes of each visual object depend only on its scene graph path. The Java 3D renderer takes advantage of this fact and renders the leaves in the order it determines to be most efficient. The Java 3D programmer normally does not have control over the rendering order of objects.

[0032] Graphic representations of a scene graph can serve as design tool and/or documentation for Java 3D programs. Scene graphs are drawn using standard graphic symbols as shown in the figure. Java 3D programs may have many more objects than those of the scene graph.

[0033] To design a Java 3D virtual universe, a scene graph is drawn using the standard set of symbols. After the design is complete, that scene graph drawing may be used as the specification for the program. After the program is complete, the same scene graph is a concise representation of the program (assuming the specification was followed). Thus, a scene graph drawn from an existing program documents the scene graph the program creates.

Rapid Processing of Scene Graph-Based Programs

[0034] First, a general overview of one embodiment of a system and method for scene graph-based programs is disclosed. Then, a more detailed explanation of one embodiment of the system and method implemented within the context of a Java 3D API is disclosed. As one skilled in the art will appreciate after reviewing this disclosure, the accompanying figures, and the attached claims, other programming languages and/or APIs may be utilized or modified to implement the system and method disclosed herein.

[0035] As noted above, To improve the flexibility and/or speed of processing scene-based programs, a graphics system (hardware and/or software) may be configured to use a scene graph directly or as a "source" structure for building various ancillary structures. A detailed description and example of the use of a scene graph is attached hereto in Attachment 2 ("Getting Started with the Java 3D™ API"). In one embodiment the graphics system may be configured to use those ancillary structure to either (i) aid in rendering and executing a scene-graph-based program, e.g., one that includes rendering (including frustum culling and occlusion culling), sound (including mono, stereo, and spatialized sounds), execution culling, collision detection, and/or picking, more efficiently; or (ii) aid in generating ancillary structures for direct execution with limited or no reference to the original scene graph (compilation).

[0036] This may include, but need not be limited to, the use of ancillary structures to reduce the extent of traversal of the original structure, replacement of the original structure by one or more ancillary structures thereby reducing references to the original structure (scene graph), or even the complete replacement of the original structure by new structures containing sufficient information to render and execute the associated scene-graph-based program (including but not limited graphics, behavior, sound, collision detection, and picking).

[0037] One particular implementation of such an ancillary structure (designed for scalability) is a lower dimensional version of k-dop hierarchy, in this case a hierarchy of axis-aligned bounding-boxes (6-dops) for use in visibility detection, frustum culling, occlusion culling, execution culling, collision detection, bounding volume determination, and picking. One such embodiment is disclosed in U.S. Patent Application Serial No. 09/247,466, filed on February 9, 1999, entitled "Visible Object Determination for Interactive Visualization."

[0038] In some embodiments, the system may support a system-based scheduler, called master control, that handles management of various resources including the resources of single and multi-processor systems. Master control may manage the various renderers and system computations and allocates their execution to a computer's various processors, memory systems, and even other network-attached computers. Master control can serialize the system's various computations or to execute those computations in parallel. It can allow adding or removing rendering and computational components both at initial startup time or dynamically over time as the system continues to execute. This orthogonalization of the various rendering and computational components allows extending the scene-graph-execution system with new renderers (such as haptics) or replacing an existing renderer with a different implementation of that renderer.

[0039] In some embodiments, the system may also implement the use of message passing for specifying state changes. These messages are labeled with a monotonic increasing time-/event-stamp. These "stamped" or labeled events allow concurrent or distributed processing while ensuring consistency.

[0040] In some embodiments, the system may be configured to use "epochs" to ensure consistency. An epoch consists of all messages having compact (continuous) set of labels (time-/event-stamps) that fall within an selected interval of "stamped" messages. The continuous declaration of an new epoch allows renderers and other computations to work within the current epoch and thus work with a consistent set of changes.

[0041] In some embodiments, an ancillary structure called the geometry structure may be used. The geometry structure may be configured to contain the current consistent state of app spatially-located renderable objects. The structure may be a bounding-box hierarchy that uses a cluster algorithm for grouping renderable, spatially-located, objects. It

NVIDIA Ex1002 Page 164 of 1081

may also include specialized method for rapidly including and excluding larger number of objects (switch processing). It may further include specialized methods for re-balancing the hierarchy to retain desirable properties. It permits consistent and concurrent access to its objects.

[0042] An ancillary structure called the rendering-environment structure that contains the current consistent state of all non-renderable spatially located objects (including non-geometric objects such as lights, fog, background, clip, model-clip, etc. nodes). It shares the specialized method for rapidly including and excluding objects and for re-balancing the hierarchy to retain desirable properties. It permits consistent and concurrent access to its objects.

[0043] In some embodiments, the graphics system may be configured o generate/support an ancillary structure called a render-bin. A render bin may be configured to contain a superset of the objects that need to be rendered for a particular frame. The render bin may "age" objects held within the structure and remove "over-age" objects during occasional "compaction" computations. The render-bin structure may be MT/MP (multithread/multiprocessor) enabled, thereby allowing concurrent access both for adding objects and for access to objects. The structure may both sort objects to be rendered into an order more compatible with the underlying rendering hardware architecture and for other local (almost "peephole") optimization that allow renderable components to combine or split for more appropriate matching to the underlying hardware. The renderable objects, held within the render bin, can have associated callbacks that will execute before the associated object is rendered, and they can hold references to multiple state change objects. They can also contain system specific computations including multiple transform matrices, rendering attributes, and levels of geometric detail, or user-specified computation versions of each of the previous. The render-bin structure may be optimized to include or exclude large groups of renderable objects (switch mode processing).

[0044] Turning now to Figure 4, a diagram illustrating one embodiment of a method for creating a parallel structure for managing and rendering scene graphs is shown. The scene graph comprises locale node 54, transform nodes 66A-D, view platform 74, behavior node 92, and light node 90, and geometry node component 96. Note that the figure only illustrates a portion of the scene graph and that additional nodes and node components (e.g., BranchGroup nodes) are present but not shown in the figure for simplicity.

[0045] As shown in the figure, as the scene graph is created or initially traversed, a parallel configuration comprising a number of structures 200-210 is generated. each structure is responsible for a particular type of object in the scene graph. For example. geometry structure 200 is responsible for the geometry node components of the scene graph. Similarly, rendering attribute structure 202 is responsible for rendering attributes such as the current viewpoint and view frustum. Rendering environment structure 204 is responsible for lighting and other environmental effects such as fog. Transform structure 206 is responsible for transformation group nodes, and behavior structure 208 is responsible for behavior nodes (e.g., changes to objects as the result of certain conditions being met, such as a collision being detected). Render bin 210 is responsible for the selecting and rendering objects.

[0046] As shown in the figure, the parallel data structures 200-208 may each have an update thread 200B-208B. The update threads may be invoked (e.g., based on messages in the corresponding message queue) to process changes. For example, assuming the color of the light node 90 changes, a message to that effect may be sent to rendering environment structure 204's message queue 204A. The rendering environment structure's update thread 204B may be invoked to check the message queue 204A for messages, detect the update message, and change the rendering environment structure 204 accordingly. In some cases, the update thread may generate its own messages that are conveyed to other structures (e.g., to render bin 210).

[0047] Render bin 210 may be configured with a number of different render threads 212A-212N in lieu of an update thread. These threads may operate in parallel and render different portions of the render bin 210. One example of this is a system utilizing a head-mounted display for virtual reality. Head mounted displays typically have two screens (one for each eye). To achieve a visible stereo effect, each display typically displays a different image, each rendered according to a different viewpoint. Thus, a different render thread may be started for each of the two different displays. Similarly, in some virtual reality applications, multiple users may each have their own viewpoint as they individually navigate the virtual world. In another embodiment, a different render bin is generated for each viewpoint (this described in greater detail below). The render threads may still operate in parallel, but may each correspond to a particular object or objects instead of a particular view.

[0048] Note, while a number of different types of parallel data structures are shown in the figure and described herein, additional types of parallel data structures may be created as necessary (e.g., for sound or force feedback device support), depending upon the exact implementation. Similarly, a number of additional threads may also be created.

[0049] Advantageously, by using this parallel structure the scene graph need be traversed only once to generate the parallel structures. By reducing the number of scene graph traversals, rendering performance may potentially be improved. Furthermore, different portions of the parallel structures and their corresponding threads may be conveyed across a network to another computer for parallel processing (i.e., to one or more non-shared memory computers). Once each structure is set up, execution may occur in parallel, with state change information being conveyed via messages transmitted between the different computers or processors. In contrast, continually repeating traversals of the scene graph does not lend itself as well to a parallel processing implementation, particular when the parallel proc-

NVIDIA Ex1002 Page 165 of 1081

essors do not share memory.

[0050] Each structure 200-210 has a corresponding message queue 200A-210A that is configured to receive and store messages. As shown in the figure, during the creation or initial traversal of the scene graph, messages may be generated and conveyed to the corresponding parallel structures. For example, when geometry node component 96 is added to the scene graph, a message is sent to geometry structure 200's message queue 200A. Similarly, when light node 90 is added to the scene graph, a message is sent to rendering environment structure 204's message queue 204B.

[0051] Turning now to Figure 5, a flowchart of one embodiment of a method for creating parallel structures for managing and rendering scene graph-based graphics data is shown. In this embodiment, the method begins (step 100) with the creation of a universe (step 102). Next, children nodes (e.g. as shown in figures 3 and 4) and branch graphs are created (step 104). This process continues until the tree is complete (step 106). Next, the tree is traversed (step 110). For each node detected (step 112), a message (e.g., "InsertNode") is generated (step 114). Similarly, for each new view detected (step 116), created a message (e.g., "CreateRenderBin") is generated (step 118). This process continues until the tree has been completely traversed (step 120). Note, in other embodiments the messages may be generated concurrently with the creation of the scene graph tree.

[0052] Next, a time counter may be initialized (step 122), and the master control thread may be started (step 130). The master control thread is configured to define the current epoch (step 132), and then schedule threads for execution (step 134). This process continues until execution is complete (steps 136-138). An epoch is a time period defined by the master control thread to allow for efficient scheduling of thread execution. Messages may be timestamped, thereby allowing the master control thread to schedule threads based on the age (and importance) of the messages or tasks to be processed by the threads. One embodiment of this time management process is described in greater detail below.

Example Embodiment Using Java 3D™ API

Java 3D Architecture

[0053] The Java 3D API may be used to implement one embodiment of the system and method described above. One such implementation is described in detail below. However, it is noted that other implementations are also possible and contemplated. To describe this particular embodiment, an overview of the Scene Graph Structure will be presented, and then components of a runtime system will be presented. Next, the process of system startup, runtime system operation, and system shutdown are presented. Then, specific features of this embodiment are explained in greater detail. Also note, that standard Java™ and Java 3D variable names, object names, and processes are used herein. For more information on Java and Java 3D, the reader is directed to the books titled "The Java 3D™ API Specification," by Sowizrat, Rushforth, and Deering, published by Addison-Wesley Publishing Co., ISBN: 0201710412, and "The Java™ Programming Language" by Ken Arnold, James Gosling, David Holmes, published by Addison-Wesley Publishing Co., ISBN: 0201704331.

Scene Graph Structure

[0054] The Java 3D API is a set of Java classes. Applications create instances of these classes and invoke their methods to control Java 3D. In one implementation, many of these classes may be merely wrapper classes around retained classes that actually handle the implementation. When an API object is created, that object may create its retained counterpart. This is true for all classes that are subclasses of Node and NodeComponent. In one embodiment, the working scene graph structure is implemented on the retained side alone.

[0055] Turning now to Figure 6, one embodiment of the scene graph structure is shown. In this embodiment, the scene graph structure and all the child/parent relationships are implemented in the retained classes. The API classes may be configured as wrapper classes (54, 62, 66A, 66B, 64A, 64B) around the retained classes (62', 66A', 66B', 64A', and 64B'). In addition, all scene graph links (e.g., parent/child relationships) are implemented on the retained side. In some embodiments most of each class may be implemented on the retained side. As also shown in the figure, in this embodiment the "Locale" class has no parallel retained class. One possible advantage of using this type of internal structure is that capability bit processing and scene graph compilation are simplified (as described in greater detail below).

Capability Bit Processing

[0056] Capability bits may be used in the Java 3D API by applications to inform Java 3D what modifications or queries the application will make to different parts of the scene graph. Each Node and NodeComponent in the scene graph may have capability bits defined for that Node or NodeComponent. After a branch graph has been made live (inserted

NVIDIA Ex1002 Page 166 of 1081

into the scene graph) or compiled (via BranchGroup.compile() or SharedGroup.compile()), the application may only modify or make queries based on the capability bits that it has set for each object. This may be implemented by having the methods in each of the API classes responsible for checking capability bits before calling the parallel retained method.

[0057] The following code is an example of such a test from the Material class:

```
public final void setAmbientColor(Color3f color) {
if (isLiveOrCompiled())
if (!this.capabilities.get(ALLOW_COMPONENT_WRITE))
throw new CapabilityNotSetException("Material: no capability
to set component");
((MaterialRetained)this.retained).setAmbientColor(color);
}
```

[0058] The Material.ALLOW _COMPONENT_WRITE capability bit is the one defined to control the setting of the material attribute. Some methods have been defined to not be modifiable once the Object has become compiled or live. The following code is an example from ImageComponent2D:

```
public final void set(BufferedImage image) {
checkForLiveOrCompiled();
((ImageComponent2DRetained)this.retained).set(image);
}
```

[0059] In this case, the checkForLiveOrCompiled() will throw an exception if the component is live or compiled. In some embodiments, all Nodes and NodeComponents may enforce capability bit processing. Therefore, all Nodes and NodeComponents have parallel retained classes. As changes are made to the API, Nodes and NodeComponents may be given new features that may require new capability bits. To facilitate this while at the same time trying to maintain a compact overall set (i.e., so that the capability bit test can be performed quickly), there is a class may be included that defines all the capability bit values. This class may assign the bit values in linear order based on the class hierarchy. Advantageously, any path in the class hierarchy may have bit values progressing in linear order. In addition, independent hierarchies can have the same bit values, thereby providing a compact set of bit values.

[0060] In Java 3D, the deepest class hierarchy is the ConeSound hierarchy, with a maximum capability bit value of 39. Some implementations may set a limit or goal of less than 64 bits (i.e., the number of bits available in a "long" variable type).

Compiling Branch Graphs

[0061] The Java 3D API allows applications to compile a BranchGraph or SharedGraph (via BranchGroup.compile () and SharedGroup.compile()). At compile time, the implementation may analyze the subgraph and perform optimizations on the subgraph. The implementation may use the capability bits that the application has set to guide the optimizations. An example of one possible optimization is to flatten the scene graph.

[0062] Turning now to Figure 7, one embodiment of a method for flattening a scene graph is illustrated. The figure illustrates two features. First, in this example the TransformGroup Nodes 66A-B have no capability bits set, so they may not be modified or queried. As a result, when the branch graph was compiled, the TransformGroupRetained Nodes were eliminated with their composite value cached in the Shape3DRetained node 64".

[0063] Second, one possible advantage of configuring the API classes as wrapper classes around the retained classes is shown. Since the TransformGroup nodes have no capability bits set and this branch has been compiled, there is no way for the application to modify or query the TransformGroupRetained nodes. Thus, they can be eliminated. The application still may have references to the TransformGroup nodes, but since no capability bits are set, they do not matter.

[0064] Thus, the figure illustrates one example of compilation optimization. In general, however, compilation optimi-

NVIDIA Ex1002 Page 167 of 1081

zations are independent from optimizations in the rest of the Java 3D implementation. In some embodiments, the optimizations may be performed before the Java 3D runtime system ever receives or processes the subgraph. Other compile optimizations may include clustering small Shape3D nodes, splitting large Shape3D nodes, stripifying geometry data, and even compressing geometry data.

Runtime System Overview

[0065] The Java 3D API is configured to support a multithreaded fully asynchronous implementation. The API also allows an implementation to use either the Scene Graph as its working structure, or alternate data structures to improve performance. In some embodiments, the Scene Graph is the main working structure. For many features, this is a useful working structure. However, when possible optimizations are considered, the scene graph may actually be a limiting structure to work with. Thus, in other embodiments a more flexible architecture may be supported to provide for greater optimizations. In one embodiment, the more flexible architecture includes three components: structures, threads, and messages. Each of these components is described in greater detail below.

Structures

[0066] A structure is an object that manages a collection of other objects. Structures are only dependent on the contents of the Scene Graph. They are responsible for optimizing the objects that they contain, given the set of requirements on that structure. They are independent of the scene graph structure and are only responsible for their own objects. The parent class of all structures is J3dStructure. Listed below are descriptions for the different types of structures that may be supported in one embodiment of the system.

GeometryStructure

[0067] The GeometryStructure is responsible for organizing all geometry and bounds data in the scene graph. There is one GeometryStructure per VirtualUniverse.

BehaviorStructure

[0068] The BehaviorStructure is responsible for organizing all Behavior nodes. It is not responsible for implementing behavior scheduler semantics. It only organizes the nodes and their wakeup criteria for fast execution culling and processing. There is one BehaviorStructure per VirtualUniverse.

SoundStructure

[0069] The SoundStructure is responsible for organizing all Sound nodes. It is not responsible for implementing sound scheduler semantics. It only organizes the nodes for fast sound render culling. There is one SoundStructure per VirtualUniverse.

RenderingEnvironmentStructure

[0070] The RenderingEnvironmentStructure is responsible for organizing objects that effect the rendering environment. This includes Light, Fog, Clip, Background, and ModelClip nodes. It does not implement their semantics. It only provides fast methods for determining the environment characteristics that apply to a scene or particular geometry. There is one RenderingEnvironmentStructure per VirtualUniverse..1.3.1.5

RenderingAttributesStructure

[0071] The RenderingAttributesStructure is responsible for managing all rendering NodeComponent objects and their mirror objects. There is one RenderingAttributesStructure for all of Java 3D.

RenderBin

[0072] The RenderBin structure is responsible for organizing the collection of visible geometry objects optimally for rendering. There is one RenderBin per View.

NVIDIA Ex1002 Page 168 of 1081

TransformStructure

[0073] The TransformStructure is responsible for updating all transforms and bounds used within the system.

Threads

[0074] Threads are configured to use the structures described above. There are many different types of threads in the 1.2 implementation. This section presents all the thread types in the system.

MasterControl

[0075] MasterControl is the controlling thread for the system. It does scheduling for all other threads in the system. It is also responsible for taking time snapshots. This will be covered in the Time section. There is one MasterControl for the entire system.

J3dThread

[0076] J3dThread is the abstract parent class for all other threads in the system. It implements all the functionality for initializing, being notified by MasterControl, and cleanup for the threads. All threads in the system, aside from MasterControl, are a subclass of J3dThread.

StructureUpdateThread

[0077] Every structure has a StructureUpdateThread. When messages are sent to a structure, the StructureUpdateThread is scheduled to retrieve those messages for its structure. It only retrieves messages for the structure that fall between the time interval for that update pass. This is described in greater detail below.

Renderer

[0078] This is the thread that renders the geometry. It also issues the SwapBuffers command. There is one Renderer per Screen3D. When there are multiple Canvas3Ds on a single screen, the Renderer thread is run for each Canvas3D. The Renderer uses the RenderBin and the RenderingAttributesStructure.

BehaviorScheduler

[0079] This thread is responsible for scheduling and processing active behaviors that have triggered. It uses the BehaviorStructure. There is currently one BehaviorScheduler per VirtualUniverse.

SoundScheduler

[0080] This thread is responsible for sound scheduling and sending sound rendering requests to the AudioDevice3D. There is one SoundScheduler per View. It uses the SoundStructure.

InputDeviceScheduler

[0081] This thread is responsible for scheduling non-blocking input devices. There is one InputDeviceScheduler per PhysicalEnvironment.

Messages

[0082] Messages are used to communicate any change that happens in the system, internal or external, through the system. In one embodiment, a single message class is implemented called "J3dMessage". Each message may have the following fields:

time -	The time when the message was sent. This is filled in by MasterControl, who is the keeper of time.
refcount -	The reference count of this message. A message could go to multiple destinations. When all destinations consume the message, its refcount goes to 0, and the message gets put on the freelist.

NVIDIA Ex1002 Page 169 of 1081

(continued)

threads -	A bitmask representing all thread types that need this message.
universe -	The universe where this message originated.
view -	The View that should receive this message. If it is null, all Views get the message.
type -	The type of this message.
args -	An array of 5 object references. Arguments are passed to the destinations through this field.

[0083] MasterControl keeps a freelist of messages, so when a message is needed, GC activity is reduced.

Time

[0084] As used herein, references to time do not refer to the time measured in System.currentTimeMillis(). Instead, it is more accurately called the change count. MasterControl has a field called "time" that starts out as 0, and is incremented (i) each time a message is sent into the system, and (ii) when MasterControl takes a time snapshot. This concept of time is used to ensure that the various threads are performing their tasks on data snapshots that are consistent across all threads. This is described in greater detail below (see section heading MasterControl).

System Startup

[0085] Bootstrapping Java 3D includes performing some system-related operations in order to prepare the system for processing. In one embodiment, this process only occurs once per invocation of the Java 3D system. The bootstrapping code is contained in MasterControl. MasterControl is a static member of the VirtualUniverse class, which means that only one MasterControl will exist in the Java 3D system. Therefore, the constructor for MasterControl is used to execute bootstrapping code. For example, the type of native graphics library to be used may be queried, and then loaded.

[0086] The class that contains the native rendering API to use is called NativeAPIInfo. A different implementation of this class may be compiled for each different target platform. This class has one method, getRenderingAPI(), and its possible values are RENDER_OPENGL_SOLARIS, RENDER_OPENGL_WIN32, or RENDER_DIRECT3D. Note, however, that the selection of which rendering library to use may be compiled into the system or it may be made runtime selectable by expanding the bootstrapping process.

[0087] Applications may be configured to create Java 3D objects in any order. However, there may also be conditions that cause Java 3D to bootstrap itself. First, as seen above, if a VirtualUniverse is created, bootstrapping will occur. There are other cases, like Canvas3D, where native library access is needed before a VirtualUniverse is created. In these cases, a static initializer method may be included in the class. That method may be configured to call the static method VirtualUniverse.loadLibraries(), which will cause the static MasterControl to get created, and bootstrapping will occur.

Universe Structures

[0088] A number of the structures are created when a VirtualUniverse is created. These include the GeometryStructure, BehaviorStructure, SoundStructure, and RenderingEnvironmentStructure. They are created here because they are View/Rendering independent. They need to be available when Nodes are added to the scene graph, even though a view may not be active yet. The RenderingAttributesStructure is created when MasterControl is created because there is only one for all of Java 3D.

Canvas3D Creation

[0089] To create a Canvas3D, applications must first get a GraphicsConfiguration that meets their needs. This involves filling out a GraphicsConfigTemplate3D object and passing that object to the getBestConfiguration() method on the device to be rendered. In the current implementation only the default screen device is supported. GraphicsDevice.getBestConfiguration() simple calls the GraphicsConfigTemplate3D.getBestConfiguration() method passing all available configurations as its argument. It is then the responsibility of the GraphicsConfigTemplate3D class to select the best GraphicsConfiguration for that template and return it. GraphicsConfigTemplate3D is a Java 3D class, so it has control over the selection process. In GraphicsConfigTemplate3D, there is a class, NativeConfigTemplate3D, that is built into Java 3D at compile time which implements the platform/rendering API specific selection semantics. For Solaris, where OpenGL is the only rendering API, the list of GraphicsConfigurations passed in is ignored, and the selection is

NVIDIA Ex1002 Page 170 of 1081

performed in native code by doing a glXChooseVisual(). Once the visual is selected, its id is passed back and matched with the GraphicsConfigurations passed in. The GraphicsConfiguration that matches is returned. On win32, for both OpenGL and Direct3D, the selection process does nothing. The PixelFormat for a win32 window may be changed at any time, so Java 3D simply changes the PixelFormat of the window when the first rendering GraphicsContext is needed by the system This will be covered more when the Renderer is covered.

[0090] At the end of this, the Canvas3D is capable of being rendered into with the current rendering API. However, the system does not start up until the Canvas3D is ready to be rendered into, or activated. This process is described below.

View Activation

[0091] The runtime system is started in two steps. The first one is triggered by a Canvas3D object being added to an AWT (abstract window toolkit) container (i.e., when Canvas3D.addNotify() is called). In response, the canvas notifies its View that it has been added by calling View.addNotify(). At this point, the View registers itself and the provoking canvas with MasterControl. Next, the Views RenderBin and SoundScheduler are created and initialized. If this is the first view to be registered with MasterControl, then the BehaviorScheduler is also created and initialized. If this is the first canvas registered for its screen, the Renderer for that screen is also created and initialized.

[0092] At this point, all threads and structures have been created and initialized. The View, however, is still not active (i.e., its structures will not receive messages and its threads will not be scheduled by MasterControl). These will not occur until a Canvas3D associated with the view is activated. In one embodiment, three conditions must be satisfied before a Canvas3D may be activated. First, a paint event must have been received on the Canvas3D. This is a requirement based on how AWT allocates underlying window system resources for Canvas3Ds (this is described in greater detail below in the Renderer section). Second, the Canvas3D must be in a running state. This is controlled directly by the Canvas3D.start/stopRenderer() methods. The third and final condition is that the Canvas3D must be visible on the screen. This is monitored by tracking various events on the Canvas3D. To track events on the Canvas3D, an internal class (EventCatcher) implements all the appropriate listener interfaces for events that the Canvas3D or BehaviorScheduler are to track. The EventCatcher receives the events and then passes them onto the BehaviorScheduler. The EventCatcher also notifies the Canvas3D that its state has changed. If the state of any of the three conditions change, Canvas3D.evaluateActive() is called to re-evaluate the Canvas3D's state. If the Canvas3D's active state changes, it notifies the View that it is associated with to evaluate its state - via View.evaluateActive(). In one embodiment, there are two conditions that are met before it is activated. First, it must be attached to a live ViewPlatform. And second, one of the Views Canvas3Ds must be active. If the state of the View changes, it activates or deactivates itself.

[0093] View activation involves a several steps. First, a message is broadcast to the threads in the system that a new View has been activated. This is done to ensure that the new View is registered with all interested structures. This message also makes the initial visibility and render operations happen on this View. Next, the View gets an id number if it doesn't have one. The use of this id is discussed below. After receiving the id number, the View checks to see if its VirtualUniverse has a primary view. If it doesn't, then it has MasterControl select a new primary view. Then, the View has its VirtualUniverse enable all currently enabled events (as selected by the BehaviorScheduler) on all the Canvas3Ds associated with this View.

[0094] Note - ComponentEvents are always enabled for Canvas3Ds to ensure activation tracking is always done. Finally, the View has MasterControl activate it. At this point, MasterControl marks the Views RenderBin and SoundScheduler as active. If this is the only active View, MasterControl also activates the BehaviorScheduler. At this point, the system is in a normal operating state.

System Flow

[0095] Since this implementation is based on message passing, there is no set system flow that occurs. However, to explain how the components interoperate, an example application scenario can be used as an example. In this example, the application is HelloUniverse, a single cube spinning. From Java3D's point of view, this application consists of one Shape3D node with a TransformGroup above it that is changing. The TransformGroup is being changed from a Behavior that gets triggered once per frame. To see how this looks internally, one looks to MasterControl first.

[0096] Turning now to Figure 8, one embodiment of the threads managed by MasterControl for the HelloUniverse example application are shown. While the number of threads shown in the figure may seem high for such a simple application, only the ones with an "X" in the box will be run on each iteration. Also note that each structure and update thread are conditionally created based on what is in the scene graph. First, to start an interation, MasterControl takes a time snapshot. In other words, at this point in time, MasterControl decides which threads need to be run on this particular iteration. This decision is based on what messages have been sent to each thread. In the example shown in the figure, two update threads are run. Since the TransformGroup is changing every frame, the GeometryStructure

NVIDIA Ex1002 Page 171 of 1081

and the RenderBin need to track the position of the Shape3D node. Thus, their update threads get scheduled for each iteration. In this case, both update threads update their Transform3D values for the Shape3D node. Since the GeometryStructure and the RenderBin are working on two different reference times, the values they cache are different. This is accomplished with a new object called the CachedTransform. This object is explained in greater detail below in connection with TransformGroup nodes. MasterControl waits for all update threads to finish before proceeding with this iteration. This ensures that the rest of the threads work with valid data without having to track state updates.

[0097] The figure actually shows two thread lists. The first one contains update and "work" threads, while the second list has render threads. MasterControl keeps two lists so that it may process them in parallel, giving the render threads a higher priority. The render threads always work off of the RenderBin structure, so there may be some constraints on what the RenderBin may do while processing messages. This is described in greater detail in the RenderBin section below. In the HelloUniverse example, the first Render thread will be scheduled first. It renders the RenderBin. At the same time, each update thread is allowed to run. In this example, that includes the TransformStructure and the GeometryStructure.

[0098] Next, the BehaviorScheduler is allowed to run, and finally the RenderBin update is allowed to run. During this time, if the Render thread finishes, the render Swap thread is allowed to run. This completes an iteration of HelloUniverse.

[0099] Next, a brief description of what each thread does in this example is provided. Since a TransformGroup is changing each frame, the TransformStructure needs to update the localToVworld values and bounds in virtual world coordinates for geometry objects in the system. All update threads after the TransformStructure cannot be scheduled until after the TransformStructure update completes. This is because most structures rely on its updated values.

[0100] The next structure to be updated is the GeometryStructure. This structure maintains a spatial data structure of all geometry data in the scene. If one object moves, then this data structure is updated to reflect the new position of the object.

[0101] The BehaviorScheduler and RenderBin update may execute at the same time. The BehaviorScheduler will execute behaviors that have been triggered since the last cycle. In this case, it is the RotationInterpolator that is spinning the cube.

[0102] For the RenderBin update, since the cube moved the RenderBin checks if the same lights and fog apply to this geometry. If the same ones apply, then it is done - which is the case here. The RenderBin update thread may not actually modify the RenderBin. This is because the Renderer(s) are executing the RenderBin at the same time as the RenderBin update thread. If a change is needed, it is simply recorded and is deferred until the ObjectUpdate time (described in greater detail below).

[0103] In parallel with this, the Renderer threads may also be running. The first Renderer thread invocation is responsible for rendering all visible objects into the Canvas3D. Once it has completed, the second Renderer thread invocation will perform the buffer swap. After the buffer swap has been completed, MasterControl may be configured to notify the BehaviorScheduler that a frame has elapsed. That notification triggers the behavior again, and the process continues.

System Exit

[0104] One potential advantage of the Java memory model is that system exit and cleanup is relatively simple. If the application removes all references to its Java 3D objects, and if all Java 3D threads exit, then the Java garbage collector will reclaim all of the Java resources that Java 3D has allocated. Unfortunately, in some implementations there are is not support for notifying Java 3D when the application is finished with Java 3D. Thus, an alternative mechanism may be used to ensure that Java 3D threads exit when they are not needed.

[0105] In one embodiment, this mechanism may be implemented in the existing addNotify/removeNotify scheme. For example, as Canvas3Ds are removed from their containers, they may be configured to notify their Views. When a View is removed, it notifies MasterControl via MasterControl.unregisterView(). MasterControl then stops all the threads associated with that View. If that was the last registered View, then MasterControl stops the VirtualUniverse based threads, and finally itself. Since this removes all active Java 3D threads, when the user no longer has any Java 3D references, it will be completely reclaimed by the JVM (Java Virtual Machine). This completes the overview of the Java 3D runtime system. Below, individual components that may be used to implement the system and method described herein, their architecture, and the algorithms they use are described in greater detail.

MasterControl

[0106] MasterControl is the global controlling object in Java 3D. It is a static object in the VirtualUniverse class, so there is only one in the system. Its main function is to perform user-based scheduling of all other threads. It may also implement a few of the API features.

NVIDIA Ex1002 Page 172 of 1081

Thread Creation and Destruction

[0107] In one embodiment, all schedulable threads in the system are subclasses of J3dThread. This class implements all initialization, communication, and destruction code for the threads. This gives MasterControl a common interface for creating, scheduling, and destroying threads.

[0108] There are two types of thread associations in the system, i.e., threads associated with a VirtualUniverse and threads associated with a View. Each will be discussed in turn. Regarding VirtualUniverse threads, the Structures for a VirtualUniverse are created when the VirtualUniverse is created. The threads associated with a VirtualUniverse only need to be initialized when there is a registered view in that VirtualUniverse. So, that becomes the triggering event for initializing those threads. When the first view of a VirtualUniverse is registered, MasterControl initializes the threads for the VirtualUniverse. The structures that get created and initializes are the GeometryStructure, BehaviorStructure, SoundStructure, and RenderingEnvironmentStructure. Again, the RenderingAttributesStructure is created by MasterControl as one global structure. Each of these structures has an update thread that gets registered with MasterControl. Along with these structures, MasterControl also creates and initializes the BehaviorScheduler for the VirtualUniverse. Finally, if there is no MasterControlThread, this causes the MasterControCThread to be created and initialized.

[0109] For View-based threads, the triggering event is the registration of a View. When a View is registered, the threads for that View are created and initialized. These threads include the SoundScheduler and the update thread for the RenderBin. The InputDeviceScheduler for the PhysicalEnvironment of this View are also created and initialized (if needed). Also, each unique Screen3D associated with each Canvas3D of the View will have its Renderer thread initialized. Although the threads have been created and initialized, they are not active yet. This happens once a View becomes active. When a View becomes active, all the threads associated with it are activated. It is at this point that MasterControl considers them for scheduling. When a View gets deactivated, the threads are simply marked as not active. They are not destroyed at this point. This allows for stopping the threads on a View without having to destroy them.

[0110] Thread destruction for View-based threads is triggered by the unregistering of a View. Since MasterControl is completely asynchronous with View registration, some bookkeeping in needed. When a View is unregistered, MasterControl waits until a safe point in time before destroying the threads. Instead of destroying them immediately, it simply puts them on a zombie thread list. Then, when the MasterControlThread looks for changes in the thread list, it will destroy zombie threads by calling J3dThread.finish(). Since a View could possibly unregister and reregister itself before the MasterControlThread can clean the zombie list, it is desirable to have all thread creation code first check if the thread to be created is on the zombie list. Destruction of VirtualUniverse threads is triggered by removal of the last View for a the VirtualUniverse. This is an asynchronous event as well, so the threads are also put on a zombie list to be purged by MasterControl at a safe time.

MasterControlThread

[0111] The MasterControl object is split into two parts to allow for Java 3D to release associated resources when they are no longer needed. All the data is held in the MasterControl object itself, while the controlling thread is encapsulated into the MasterControlThread (MCT). This allows the system to quickly create and destroy the controlling thread while maintaining system state. Using the flow of the MasterControlThread, the bulk of MasterControl can be explained. First, the MCT checks to see if there are any tasks (i.e., work) to be performed. The Boolean variable workToDo is used to signal the MCT that there is pending work. If there are no tasks to perform, then the MasterControlThread enters a wait state until a task arrives. The work-causing action will notify the MCT. This allows Java 3D to refrain from consuming CPU resources when there are no pending tasks to be performed.

workThreads list

[0112] Next, the MCT checks to see if any new threads have been added to the system. This is done in updateThreadLists(). When a thread is added or removed from the system, the MCT updates its list of schedulable threads. When a thread change is needed, the Boolean variable threadListsChanged is set. If this is the case, then updateThreadLists() recomputes the work-Threads array of threads and cleans up any zombie threads. Next, updateWorkThreads() makes thread scheduling decisions. As part of this process, updateWorkThreads() first calculate how many entries are needed in the workThreads array. In one embodiment, the workThreads array is two-dimensional. As noted above, there is an array of update/work threads and an array of render threads. The number of entries for each is calculated as follows:

[0113] For the update/work list, the number of thread entries is:

- The number of update threads, plus

NVIDIA Ex1002 Page 173 of 1081

- The number of VirtualUniverses (one BehaviorScheduler for each), plus
- For each View:

    i. One SoundScheduler; and
    ii. The number of unique PhysicalEnvironment's (InputDeviceScheduler)

This gives the total length of the update/work array.

[0114] For the render list, the number of thread entries is:

- One render request thread, plus (For Offscreen and Immediate Mode)
- For each View:

    i. The number of Canvas3D's (1 Renderer per Canvas3D), plus
    ii. The number of Screen3D's (1 Swap per Screen3D)

[0115] The work Threads array is an array of J3dThreadData objects. The J3dThreadData object encapsulates all the information needed to run a thread for a single invocation. The data that J3dThreadData keeps includes:

- The thread reference, a J3dThread
- A set of four time values (explained in the Time section)
- A bitmask of options for this thread invocation
- The arguments to be passed to the thread
- Whether or not this thread needs to be run on this pass; and
- Arguments used by the Renderer thread

[0116] The J3dThreadData object is used to allow a thread to exist in the list of workThreads more than once. An example of this is the Renderer thread, which may be run many times for a single MCT pass. Each thread is responsible for managing its list of J3dThreadDara objects. In some embodiments, only the Renderer thread has more than one J3dThreadData object. It uses one J3dThreadData object for each View/Canvas3D pair for Rendering and one J3dThreadData object for each View/Screen3D pair for Swapping.

[0117] After the workThreads array has been allocated, updateWorkThreads() may be configured to fill in the array. As the J3dThreadData objects are added to the array, they have their various fields filled in. There are some thread run options that are used by the MCT to know how to manage the thread invocation. Some options are exclusive of other, while some may be combined.

[0118] In one embodiment the list of options include:

- J3dThreadData.CONT_THREAD: Other threads may run while this thread is running.
- J3dThreadData.WAIT_THIS_THREAD: Wait until this thread is done before allowing any other threads to run.
- J3dThreadData.WAIT_ALL_THREADS: Wait for all currently running threads to complete before allowing any other threads to run.

One of the next two flags may be combined with the above flags.

- J3dThreadData.START_TIMER: Start timing the frame associated with the given View.
- J3dThreadData.STOP_TIMER: Stop timing the frame associated with the given view. Store the data in the View.

[0119] In filling the state array, the first thread in the array is the RenderingAttributeStructwe update thread. It is first because there is only one for the whole system. Next, each universe's TransfomtStructure update thread is added. These need to be run early so that other threads may use their updated values. This completes section one of the update list. All threads in this section may run in parallel, but the update list does not continue until they are done. The next section of threads is a completion of the universe based update threads. So, for each universe, and update thread is added for the GeometryStructure, BehaviorStructure, RenderingEnvironmentStructure, and SoundStructure. All threads in this section may run in parallel, but no more update/work threads are run until they complete. This is because they will rely on the updates to these structures. The final set of update/work threads are a BehaviorScheduler for each universe, an InputDeviceScheduler for each PhysicalEnvironment, and for each View a RenderBin update thread and a SoundScheduler. These threads may also run in parallel.

[0120] The second dimension of the workThreads array is used for render threads. MCT allows parallel rendering to Canvas3D's that are on separate Screen3D's but that share the same View. To facilitate this, some data structures

NVIDIA Ex1002 Page 174 of 1081

may be updated in the View when Canvas3D's are added and removed from a View. They consist of two ArrayLists. The first, the screenList, is a list of all Screen3Ds currently active. The second, the canvasList, is a list of ArrayLists. Each ArrayList in the list is a list of all Canvas3Ds on that Screen3D.

[0121] Turning now to Figure 9, this structure is illustrated for three common cases. The first case is most common, having one Canvas3D (308) on one Screen3D (302). The second case has two Canvas3D's (308 and 320) on one Screen3D (302). The third case is two Canvas3D's (308 and 320) on two Screen3D's (302 and 316). The third case is one in which the system can render to the two canvases in parallel. The canvasList (310) is cached as a two dimensional array (310A-310B) of Canvas3D's, so it may be parsed quickly.

[0122] Returning to updateWorkThreads(), the Renderers that need to be scheduled can be derived from this two dimensional array. Each entry in the canvasList is an array of Canvas3D's for that Screen3D. The length of the canvasList array is the total number of Renderers that may run in parallel. The longest array in canvasList is the total number of parallel Render groups that need to be scheduled. Since each Screen3D may have a different number of Canvas3D's on it, each parallel render group may not be the same size. All Render threads in a parallel render group, except the last one, have their thread options set to J3dThreadData.CONT_THREAD. The last one in the render group has its flag set to J3dThreadData.WAIT_ALL_THREADS, because it will wait for everyone in its group to finish. All of these Renderer invocations have two thread arguments, the operation to perform (Renderer.RENDER) and the Canvas3D to render into.

[0123] The next set of threads to be added to the workThreads render array is the Renderer threads that issue the swap buffers commands. There needs to be one thread invocation for each Screen3D in the View. This is directly gotten by iterating over canvasList again. All of these Renderer thread invocations, except the last one, gets its thread option set to J3dThreadData.CONT_THREAD since they can all run in parallel. The last Renderer gets its thread options flag set to J3dThreadData.WAIT_ALL_THREADS and J3dThreadData.STOP_TIMER since it will wait for all threads to finish and it will stop the View timer. The arguments that each Renderer receives are the operations to perform (Renderer.SWAP), the View, and the list of Canvas3D's that are to be swapped. The last entry in the render array is a render request thread. It is used to process offscreen rendering and immediate mode requests. At this point, the workThreads array has been updated. Once updateWorkThreads() has completed, updateThreadLists() processes the list of zombie threads and kills them by calling J3dThread.finish(). Now, MasterControl has a completely updated set of threads to schedule.

Time

[0124] As mentioned above, in one embodiment there are four times associated with each thread. First, there is lastUpdateTime. This is the last time that a message was sent to that thread. Each time a message is sent to a thread, this value is updated. Next is updateTime. This is a snapshot value of lastUpdateTime. The snapshot time is taken at the beginning of each MCT iteration. Finally, there are lastUpdateReferenceTime and updateReferenceTime. These times define the snapshot range of time values that are being processed on a given invocation of a thread. These are also taken at the beginning of each MCT iteration.

[0125] Returning to the flow of the MCT, once the workThreads list has been checked, MCT takes a time snapshot for the threads. This is done in updateTimeValues(). There are two time values that are saved on each iteration, currentTime and lastTime. The variable currentTime is retrieved each time updateTimeValues() is called by calling getTime (). The variable lastTime is the previous currentTime. For each snapshot, the time values between currentTime and lastTime define the current time range.

After the two time values are updated, each workThread is examined. Each thread's lastUpdateReferenceTime is set to lastTime, each thread's updateReferenceTime is set to currentTime, and each thread's updateTime is set to the thread's lastUpdateTime. If a thread's updateTime is greater than its lastUpdateReferenceTime, then it is flagged as a thread that needs to be run.

Running Threads

[0126] Once the time snapshot has happened, the MCT proceeds to run each thread. It uses runMonitor() to let each thread run while adhering to its run flags and giving render threads a higher priority. When a J3dThread is initialized, it immediately goes into its synchronized runMonitor() method and goes into a wait() state, waiting for notification from the MCT. When it is time to run a thread, the MCT goes into its own synchronized runMonitor() method and calls into the thread's runMonitor() method to notify() it and return.

[0127] At this point, the MCT can go into a number of states. If the thread run options indicate that it is to wait for this thread, the thread reference is stored, and the MCT keeps going into a wait() state until that thread's notify() indicates to the MCT that it is done. If the thread run options indicate that the MCT should wait for all threads to complete, the MCT repeated enters into a wait() state until all running threads are done. This is tracked by having a threadPending

NVIDIA Ex1002 Page 175 of 1081

counter which represents the number of threads currently running. If neither of these options are set, the MCT may be configured to check to see if it is out of CPU resources. It does this by checking threadPending against the cpuLimit. If they are equal, the MCT goes into a wait() state until a thread completes. If none of these conditions are true, then the MCT advances to the next thread. Once all the threads have been processed, the MCT process the updateObject list, and then returns for another iteration.

Update Objects

[0128] When structures are processing their updates, it sometimes becomes necessary to modify a part of the structure that the Renderers are using. The structure cannot do the update when processing messages since the Renderers are most likely running at that time. In some implementations an interface call ObjectUpdate may be defined to address this issue. The interface call may be an interface that consists of a single updateObject() method. If an object finds that it needs to make a modification to itself for which the Renderer depends, the object implements this interface. When the modification becomes necessary, object adds itself to MasterControl's object update list. When MasterControl decides that it is safe to make modifications to render dependent structures, it calls the updateObject() method for each object on its update object list. This is how the RenderingAttributesStructure, TransformStructure, RenderBin, and others process their render dependent modifications.

Frame Timer

[0129] One of the features that MasterControl implements is the View frame timing API. It does this by adding flags to the thread run options of the threads processed by the MCT. As the MCT iterates over each thread, it checks the thread's run flags. If the J3dThreadData.START_TIMER flag is set, the View is extracted from the thread arguments, the View's frameNumber is incremented, and the frameStartTime is retrieved by calling System.cutrentTimeMillis(). Then the thread is allowed to run.
If the J3dThreadData.STOP_TIMER flag is set for the thread, the thread is allowed to run, then the View is extracted from the threads arguments, the stopTime for that View is recorded by calling System.currentTimeMillis(), and the View.setFtameTimingValues() method is called. This synchronized method takes the frameNumber, startTime, and stopTime values and updates the.currentFrameNumber, currentFrameStartTime, currentFrameDuration, and frameNumbers and frameStartTimes circular buffers. When an application retrieves any of these values with the View API's, they go through synchronized methods to ensure that these values are stored and retrieved in an MT-SAFE manner.

Messages

[0130] As noted above, messages may be used to communicate information throughout the system. MasterControl is the keeper and distributor of messages in the system.

[0131] Whenever a message is needed, it is retrieved by calling VirtualUniverse.mc.getMessage(). This MasterControl method first checks to see if the message free list has any messages. If so, it recycles one of those messages. If not, it creates a new one and returns it. Once a component has a message, there are a number of fields that need to be filled in before it is sent into the system. The first field to be filled in is the universe. Next, the type of the message is filled in. All message types are found in the J3dMessage class. Then, the thread's bitmask is filled in. This is a bitmask of all thread types that need to be sent this message. All the possible thread types can be found in the J3dThread class. Finally, any arguments for the message need to be filled in. The maximum number of arguments is currently 5, but that number is arbitrary. Once, all the fields are complete, the message gets sent into the system by calling VirtualUniverse.mc.processMessage(). There are two main types of messages that can be processed. There are messages that get sent to structures as well as notify work threads. And, there are messages that are only intended to notify a thread that it needs to run. If this latter case is all that is needed, the component may simply call VirtualUniverse.mc.sendRunMessage() to notify the threads specified in the arguments.

Sending Messages

[0132] When a message is sent to ViriualUniverse.mc.processMessage(), it goes through the following steps. First, the time field of the message is filled in by getting a new time value via getTime(). Next, each work thread that the message is destined to has its lastUpdateTime value set to the message's new time value. Then, any structure that this message is meant for is sent the message by calling J3dStructure.addMessage(). Finally, MasterControl is notified that there is work to do.

NVIDIA Ex1002 Page 176 of 1081

Consuming Messages

[0133] Messages are sent to structures by MasterControl through its processMessages() method. It is the responsibility of the structure update methods to retrieve the messages. When a thread is notified to run by MasterControl, it receives two time values, lastUpdateTime and referenceTime. The time value lastUpdateTime is the last time value when a message was sent to this structure. It also gets a referenceTime, which is the reference time that the thread is to work in. For structures, this means that the update thread needs to consume all messages less than or equal to lastUpdateTime but not greater than referenceTime.

[0134] When the structures processMessages() method gets called, the structure calls getMessages() with the time range. This method returns all messages falling into that time range. A single message can go to many structures, so each message has a reference count. Each time the message is added to a structures message queue, its reference count is incremented. The last thing each structures processMessages() method performs is to call J3dMessage. decRefcount(). When the reference count goes to 0, all its fields are nulled out and it gets put on the MasterControl message freelist.

Primary Views

[0135] MasterControl is also responsible for is assigning primary views for each VirtualUniverse. There are two cases when MasterControl can be called upon to do this. When a View becomes active, it checks to see if its VirtualUniverse has a primary View. If it does not, it calls VirtualUniverse.mc.assignNewPrimaryView(). Also, when a View is deactivated, if it is the primary View for its VirtualUniverse, it calls VirtualUniverse.mc.assignNewPrimaryView(). If the VirtualUniverse passed in has a primary view, it is one that is no longer active, so it sets that Views primaryView flag to false. Then it iterates over all active Views looking for one in the given VirtualUniverse. It stops when it finds the first one. If one is found, the View's primaryView flag is set to true and the VirtualUniverse's primary view is set to the one found. If one is not found, the VirtualUniverse's primary view is set to null.

[0136] Although the embodiments above have been described in considerable detail, other versions are possible. Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications. Note the headings used herein are for organizational purposes only and are not meant to limit the description provided herein or the claims attached hereto.

Claims

1. A method comprising:

   generating a scene graph (50), wherein the scene graph comprises information representing a plurality of three-dimensional objects; and
   generating a parallel set of structures (200-210) and their corresponding threads (200B-208B) corresponding to the scene graph, wherein said generating comprises traversing the scene graph and generating the parallel set of structures and threads based on contents of the scene graph, wherein each structure comprises an object that manages selected data from the scene graph, and wherein the plurality of threads are executable to render one or more frames corresponding to the scene graph.

2. The method of claim 1, wherein the selected data includes: appearance data, geometry data, and environmental data.

3. A computer program embodied on an electronic medium, wherein the computer software program comprises a plurality of instructions configured to perform the following:

   receive a scene graph (50), wherein the scene graph comprises information representing a plurality of three-dimensional objects, wherein the information includes: appearance data, geometry data, and environmental data; and
   generate a parallel set of structures (200-210) and their corresponding threads (200B-208B) corresponding to the scene graph, wherein said generating comprises traversing the scene graph and generating the parallel set of structures and threads based on contents of the scene graph, wherein each structure comprises an object that manages selected data from the scene graph, and wherein the plurality of threads are executable to render one or mere frames corresponding to the scene graph.

NVIDIA Ex1002 Page 177 of 1081

4. The computer program or method of claims 1-3, wherein the threads are configured to generate messages to specify state changes.

5. The computer program or method of claims 1-4, wherein the messages can be multicast to multiple structures or unicast to a single structure.

6. The computer program or method of claims 1-5, wherein each structure has a corresponding update thread configured to opdate the structure.

7. The computer program or method of claims 1-6, wherein the scene graph includes behavior data (208) for the three-dimensional objects.

8. The computer program or method of claims 1-7, wherein the environmental data (204) includes lighting and fog information.

9. The computer program or method of claims 1-8, wherein one structure is a render bin (210) configured to receive messages that store references to particular geometry data that is to be rendered in the render bin.

10. The computer program or method of claims 1-9, wherein the render bin has one or more render threads associated with it, wherein each render thread is configured to render the geometry data.

11. The computer program or method of claim 1-10, wherein the program is configured to have each structure optimize the structure's data from the scene graph.

12. The computer program or method of claims 1-11, wherein the program is configured to optimize the structure's data by stripifying geometry data.

13. The computer program or method of claims 1-12, wherein the program is configured to optimize the structure's data by splitting nodes.

14. The computer program or method of claims 1-13, wherein the program is configured to optimize the structures by flattening the scene graph.

15. The computer program or method of claims 1-14, wherein one of the structures is a bounding box.

16. The computer program or method of claims 1-15, wherein the messages include a time stamp.

17. The computer program or method of claim 16, wherein the time stamp is a change count.

18. The computer program or method of claims 1-17, wherein the messages include an event stamp.

19. The computer program or method of claims 1-18, wherein the messages are utilized to allow parallel processing of individual threads.

20. A method comprising:

receiving a scene graph (50); and
generating a parallel set of structures (200-210) and their corresponding threads (200B-208B) corresponding to the scene graph, wherein said generating comprises traversing the scene graph and generating the parallel set of structures and threads based on contents of the scene graph, wherein each structure comprises an object that manages selected data from the scene graph, wherein a first subset of the threads are executable to update the structures based on changes to the objects, and wherein a second subset of the threads are executable to render an image based on the structures without repeatedly traversing the scene graph.

21. An apparatus comprising:

means for generating a scene graph (50), wherein the scene graph comprises information representing a plurality of three-dimensional objects; and

NVIDIA Ex1002 Page 178 of 1081

means for generating a parallel set of structures (200-210) and their corresponding threads (200B-208B) corresponding to the scene graph, wherein said means for generating comprises means for traversing the scene graph to generate a parallel set of structures and threads based on contents of the scene graph, wherein each structure comprises an object that manages selected data from the scene graph, and wherein the plurality of threads are executable to render one or more frames corresponding to the scene graph.

Patentansprüche

1. Verfahren, das aufweist:

   das Erzeugen eines Szenegraphs (50), wobei der Szenegraph Informationen aufweist, die eine Mehrzahl von dreidimensionalen Objekten repräsentieren, und
   Erzeugen eines parallelen Satzes von Strukturen (200-210) und deren entsprechende Threads (200B-208B) entsprechend dem Szenegraph, wobei das Erzeugen das Durchlaufen des Szenegraphs und das Erzeugen des parallelen Satzes von Strukturen und Threads basierend auf den Inhalten des Szenegraphs aufweist, wobei jede Struktur ein Objekt aufweist, das ausgewählte Daten von dem Szenegraph handhabt, und wobei die Mehrzahl von Threads ausführbar sind, um ein oder mehrere Einzelbilder entsprechend dem Szenegraph darzustellen.

2. Verfahren nach Anspruch 1, bei dem die ausgewählten Daten beinhalten: Erscheinungsdaten, Geometriedaten und Umgebungsdaten.

3. Computerprogramm, das auf einem elektronischen Medium verkörpert ist, wobei das Computersoftwareprogramm eine Mehrzahl von Befehlen aufweist, die dafür ausgelegt sind, das Folgende durchzuführen:

   Empfangen eines Szenegraphs (50), wobei der Szenegraph Informationen aufweist, die eine Mehrzahl von dreidimensionalen Objekten darstellen, wobei die Informationen beinhalten:

      Erscheinungsdaten, Geometriedaten und Umgebungsdaten, und
      Erzeugen eines parallelen Satzes von Strukturen (200-210) und ihrer entsprechenden Threads (200B-208B) entsprechend dem Szenegraph, wobei das Erzeugen das Durchlaufen des Szenegraphs und das Erzeugen des parallelen Satzes von Strukturen und Threads basierend auf Inhalten des Szenegraphs aufweist, wobei jede Struktur ein Objekt aufweist, das ausgewählte Daten von dem Szenegraph handhabt, und wobei die Mehrzahl von Threads ausführbar sind, um ein oder mehrere Einzelbilder entsprechend dem Szenegraph darzustellen.

4. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 3, wobei die Threads derart ausgelegt sind, daß sie Nachrichten erzeugen, um Zustandsveränderungen zu spezifizieren.

5. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 4, wobei die Nachrichten zu mehreren Strukturen oder zu einer einzelnen Struktur übertragen werden können.

6. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 5, wobei jede Struktur einen entsprechenden Aktualisierungsthread hat, der dafür ausgelegt ist, die Struktur zu aktualisieren.

7. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 6, wobei der Szenegraph Verhaltensdaten (208) für die dreidimensionalen Objekte beinhaltet.

8. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 7, wobei die Umgebungsdaten (204) die Belichtungs- und Schleierinformation (lighting- und fog-Information) beinhalten.

9. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 8, wobei eine Struktur ein Darstellungsbehälter (210) ist, der dafür ausgelegt, Nachrichten zu empfangen, die Referenzen zu bestimmten Geometriedaten speichern, die in dem Darstellungsbehälter darzustellen sind.

10. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 9, wobei der Darstellungsbehälter einen oder mehrere Darstellungsthreads hat, die mit diesem verknüpft sind, wobei jeder Darstellungsthread dafür ausgelegt

NVIDIA Ex1002 Page 179 of 1081

ist, die Geometriedaten darzustellen.

11. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 10, wobei das Programm derart konfiguriert ist, daß jede Struktur die Daten der Struktur von dem Szenegraph optimiert.

12. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 11, wobei das Programm derart konfiguriert ist, daß es die Strukturdaten optimiert durch Streifenbildung der Geometriedaten.

13. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 12, wobei das Programm derart konfiguriert ist, daß es die Strukturdaten durch Splitknoten optimiert.

14. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 13, wobei das Programm derart konfiguriert ist, daß es die Strukturen optimiert durch Abflachen des Szenegraphs.

15. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 14, wobei eine der Strukturen eine Begrenzungsbox ist.

16. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 15, wobei die Nachrichten einen Zeitstempel beinhalten.

17. Computerprogramm oder Verfahren nach Anspruch 16, wobei der Zeitstempel ein Veränderungszähler ist.

18. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 17, wobei die Nachrichten einen Ereignisstempel beinhalten.

19. Computerprogramm oder Verfahren nach einem der Ansprüche 1 bis 18, bei dem die Nachrichten verwendet werden, um die Parallelverarbeitung von einzelnen Threads zu erlauben.

20. Verfahren, das aufweist:

Empfangen eines Szenegraphs (50),
Erzeugen eines parallelen Satzes von Strukturen (200-210) und ihrer entsprechenden Threads (200B-208B) entsprechend dem Szenegraph, wobei das Erzeugen das Durchlaufen des Szenegraphs und das Erzeugen des parallelen Satzes von Strukturen und Threads basierend auf den Inhalten des Szenegraphs aufweist, wobei jede Struktur ein Objekt aufweist, das ausgewählte Daten von dem Szenegraph handhabt, wobei eine erste Untergruppe der Threads ausführbar ist, um die Strukturen basierend auf den Veränderungen der Objekte zu aktualisieren, und wobei eine zweite Untergruppe der Threads ausführbar ist, um ein Bild basierend auf den Strukturen darzustellen, ohne wiederholt den Szenegraph zu durchlaufen.

21. Vorrichtung, die aufweist:

Einrichtungen für das Erzeugen eines Szenegraphs (50), wobei der Szenegraph eine Information aufweist, die eine Mehrzahl von dreidimensionalen Objekten darstellt, und
eine Einrichtung für das Erzeugen eines parallelen Satzes von Strukturen (200-210) und deren entsprechenden Threads (200B-208B) entsprechend dem Szenegraph, wobei die Einrichtung für das Erzeugen eine Einrichtung für das Durchlaufen des Szenegraphs aufweist, um einen parallelen Satz von Strukturen und Threads basierend auf Inhalten des Szenegraphs zu erzeugen, wobei jede Struktur ein Objekt aufweist, das ausgewählte Daten von dem Szenegraph handhabt, und wobei die Mehrzahl von Threads ausführbar sind, um ein oder mehrere Einzelbilder entsprechend dem Szenegraph darzustellen.

Revendications

1. Procédé comprenant :

la production d'un graphe de scène (50), dans lequel le graphe de scène comprend de l'information représentant une pluralité d'objets tridimensionnels ; et
la production d'un ensemble parallèle de structures (200 à 210) et de leurs unités d'exécution correspondantes

NVIDIA Ex1002 Page 180 of 1081

(200B à 208B) correspondant au graphe de scène, dans lequel ladite production comprend le parcours du graphe de scène et la production de l'ensemble parallèle de structures et des unités d'exécution en se basant sur le contenu du graphe de scène, dans lequel chaque structure comprend un objet qui gère des données choisies à partir du graphe de scène, et dans lequel la pluralité d'unités d'exécution est exécutable pour rendre une ou plusieurs vues correspondant au graphe de scène.

2. Procédé selon la revendication 1, dans lequel les données choisies comprennent : des données d'aspect, des données de géométrie et des données d'environnement.

3. Programme d'ordinateur incorporé sur un support électronique, dans lequel le programme de logiciel d'ordinateur comprend une pluralité d'instructions configurées pour exécuter ce qui suit :

recevoir un graphe de scène (50), dans lequel le graphe de scène comprend des informations représentant une pluralité d'objets tridimensionnels, dans lequel les informations incluent : des données d'aspect, des données de géométrie et des données d'environnement ; et
produire un ensemble parallèle de structures (200 à 210) et de leurs unités d'exécution correspondantes (200B à 208B) correspondant au graphe de scène, dans lequel ladite production comprend le parcours du graphe de scène et la production de l'ensemble parallèle de structures et des unités d'exécution en se basant sur le contenu du graphe de scène, dans lequel chaque structure comprend un objet qui gère des données choisies à partir du graphe de scène, et dans lequel la pluralité d'unités d'exécution est exécutable pour rendre une ou plusieurs vues correspondant au graphe de scène.

4. Programme d'ordinateur ou procédé selon les revendications 1 à 3, dans lequel les unités d'exécution sont configurées pour engendrer des messages pour spécifier des changements d'état.

5. Programme d'ordinateur ou procédé selon les revendications 1 à 4, dans lequel les messages peuvent être multidiffusés à des structures multiples ou monodiffusés à une structure unique.

6. Programme d'ordinateur ou procédé selon les revendications 1 à 5, dans lequel chaque structure possède une unité d'exécution correspondante de mise à jour configurée pour mettre à jour la structure.

7. Programme d'ordinateur ou procédé selon les revendications 1 à 6, dans lequel le graphe de scène comprend des données (208) de comportement pour les objets tridimensionnels.

8. Programme d'ordinateur ou procédé selon les revendications 1 à 7, dans lequel les données (204) d'environnement comprennent de l'information d'éclairage et de brouillard.

9. Programme d'ordinateur ou procédé selon les revendications 1 à 8, dans lequel une structure est une boîte de rendu (210) configurée pour recevoir des messages qui mémorisent des références à des données particulières de géométrie qui sont à rendre dans la boîte de rendu.

10. Programme d'ordinateur ou procédé selon les revendications 1 à 9, dans lequel la boîte de rendu possède une ou plusieurs unités de rendu qui lui sont associées, dans lequel chaque unité d'exécution de rendu est configurée pour rendre les données de géométrie.

11. Programme d'ordinateur ou procédé selon les revendications 1 à 10, dans lequel le programme est configuré pour que chaque structure optimise les données de structure provenant du graphe de scène.

12. Programme d'ordinateur ou procédé selon les revendications 1 à 11, dans lequel le programme est configuré pour optimiser les données de structure en mettant sous forme de bandes les données de géométrie.

13. Programme d'ordinateur ou procédé selon les revendications 1 à 12, dans lequel le programme est configuré pour optimiser les données de structure en divisant des noeuds.

14. Programme d'ordinateur ou procédé selon les revendications 1 à 13, dans lequel le programme est configuré pour optimiser les structures en aplanissant le graphe de scène.

15. Programme d'ordinateur ou procédé selon les revendications 1 à 14, dans lequel l'une des structures est un cadre

NVIDIA Ex1002 Page 181 of 1081

de contour.

16. Programme d'ordinateur ou procédé selon les revendications 1 à 15, dans lequel des messages contiennent un groupe date-heure.

17. Programme d'ordinateur ou procédé selon la revendication 16, dans lequel le groupe date-heure est un compte de changements.

18. Programme d'ordinateur ou procédé selon les revendications 1 à 17, dans lequel les messages contiennent une estampille d'événement.

19. Programme d'ordinateur ou procédé selon les revendications 1 à 18, dans lequel les messages sont utilisés pour permettre le traitement en parallèle d'unités d'exécution individuelles.

20. Procédé comprenant :

la réception d'un graphe de scène (50) ; et
la production d'un ensemble parallèle de structures (200 à 210) et de leurs unités d'exécution correspondantes (200B à 208B) correspondant au graphe de scène, dans lequel ladite production comprend le parcours du graphe de scène et la production de l'ensemble parallèle de structures et des unités d'exécution en se basant sur le contenu du graphe de scène, dans lequel chaque structure comprend un objet qui gère des données choisies à partir du graphe de scène, dans lequel un premier sous-ensemble des unités d'exécution est exécutable pour mettre à jour les structures en se basant sur des changements apportés aux objets, et dans lequel un second sous-ensemble des unités d'exécution est utilisable pour rendre une image en se basant sur les structures sans parcourir de manière répétée le graphe de scène.

21. Dispositif comprenant :

un moyen destiné à produire un graphe de scène (50), dans lequel le graphe de scène comprend de l'information représentant une pluralité d'objets tridimensionnels ; et
un moyen a destiné à produire un ensemble parallèle de structures (200 à 210) et leurs unités d'exécution correspondantes (200B à 208B) correspondant au graphe de scène, dans lequel ledit moyen de production comprend un moyen destiné à parcourir le graphe de scène pour produire un ensemble parallèle de structures et des unités d'exécution en se basant sur le contenu du graphe de scène, dans lequel chaque structure comprend un objet qui gère des données choisies à partir du graphe de scène, et dans lequel la pluralité d'unités d'exécution est exécutable pour rendre une ou plusieurs vues correspondant au graphe de scène.

NVIDIA Ex1002 Page 182 of 1081

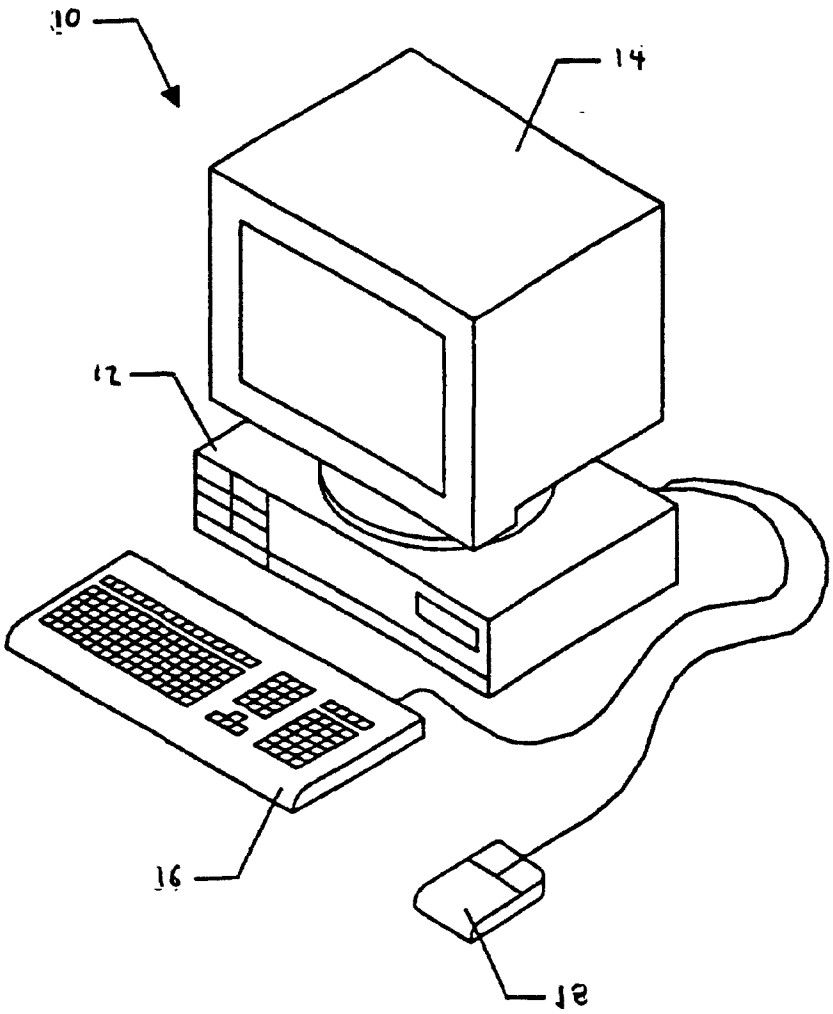


FIG. 1

NVIDIA Ex1002 Page 183 of 1081

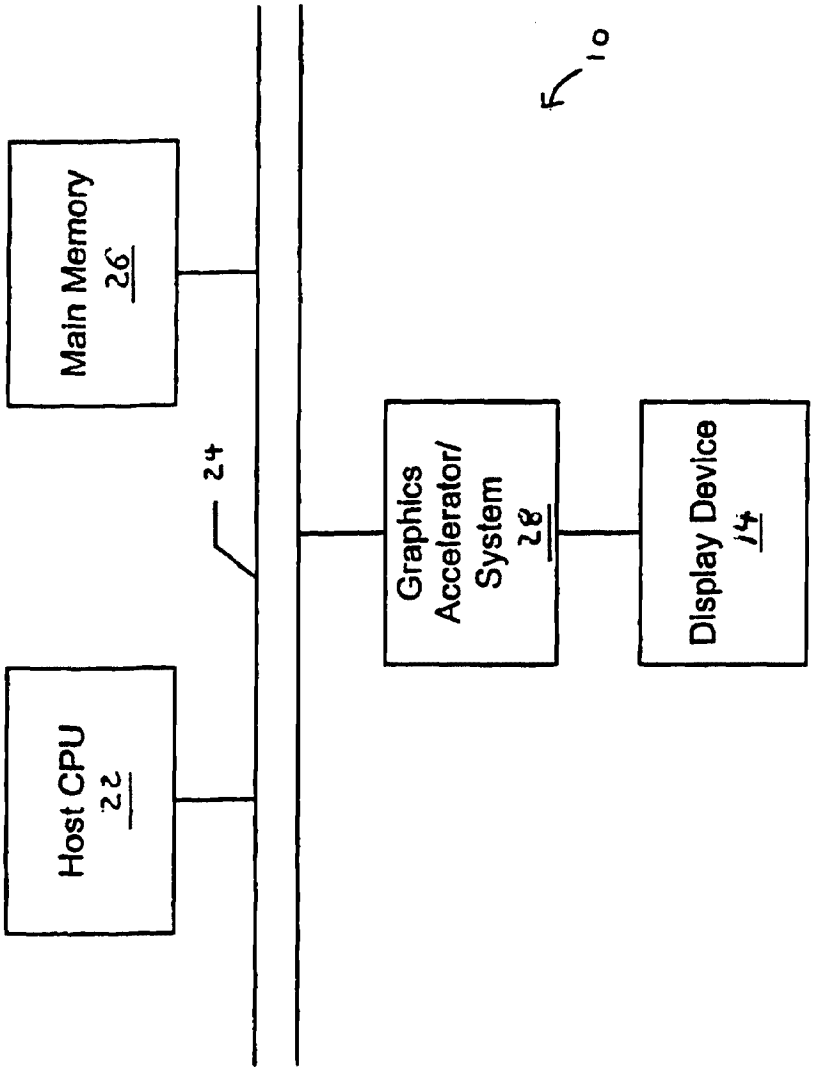


FIG. 2

NVIDIA Ex1002 Page 184 of 1081

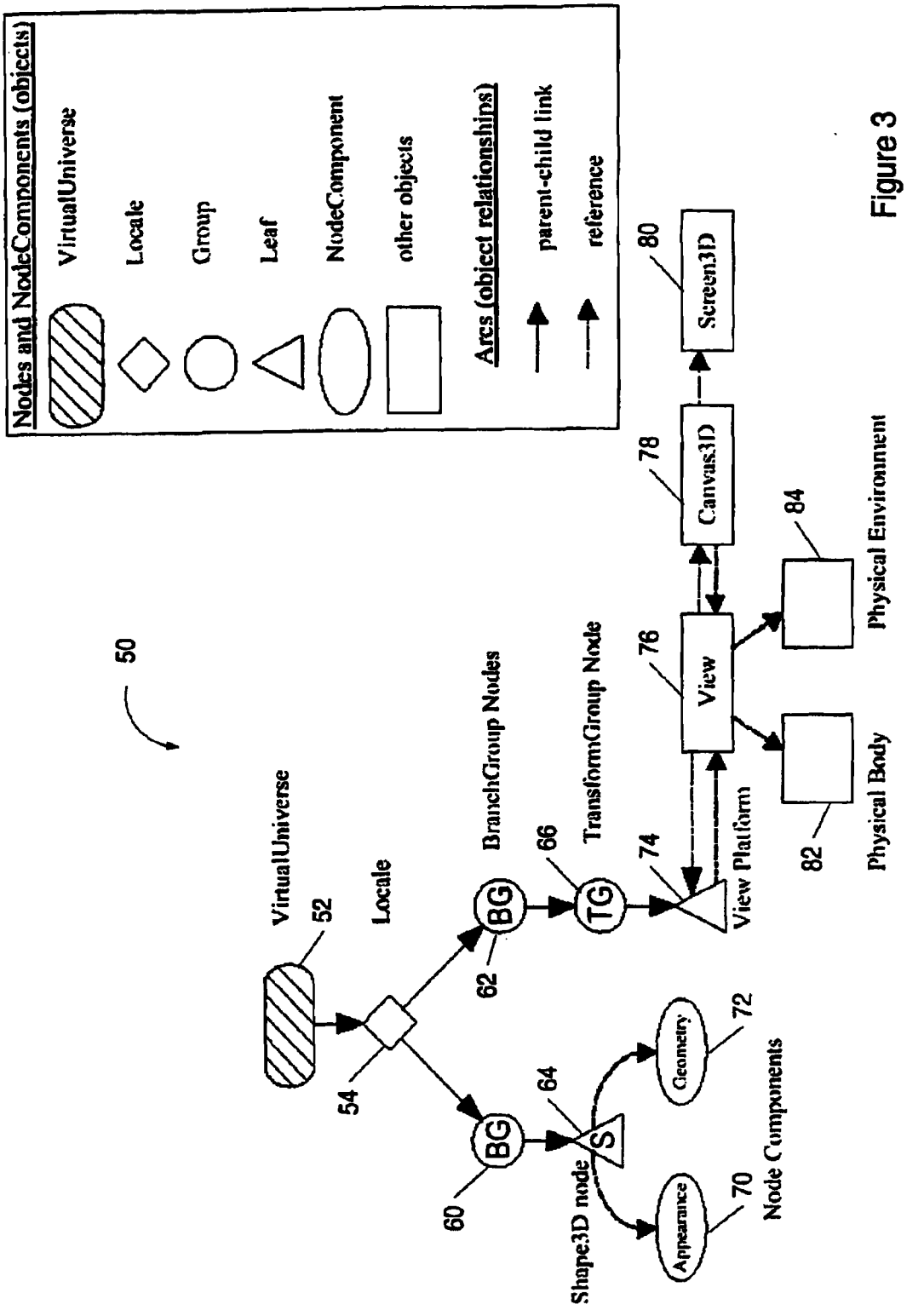


Figure 3

NVIDIA Ex1002 Page 185 of 1081

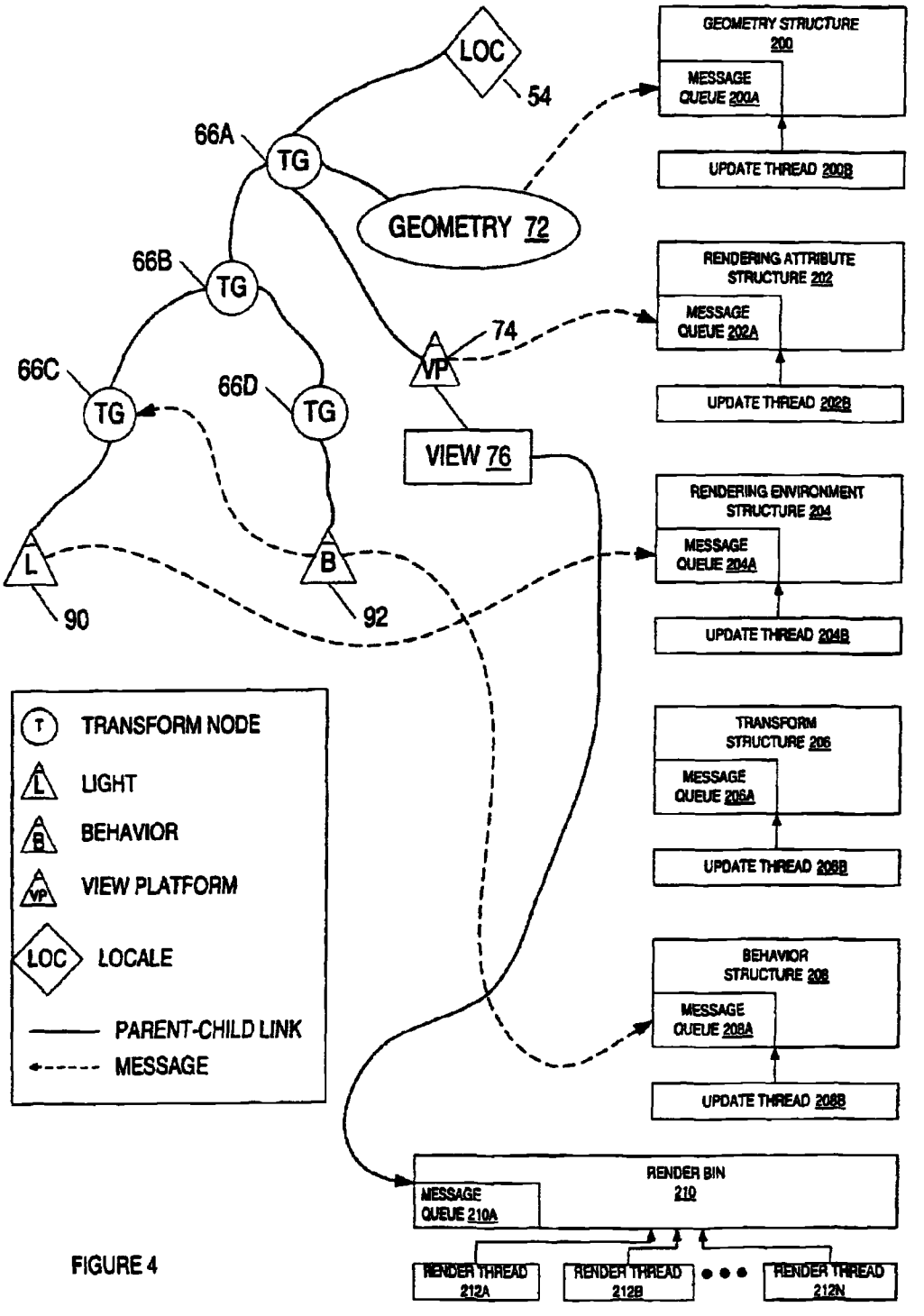


FIGURE 4

NVIDIA Ex1002 Page 186 of 1081

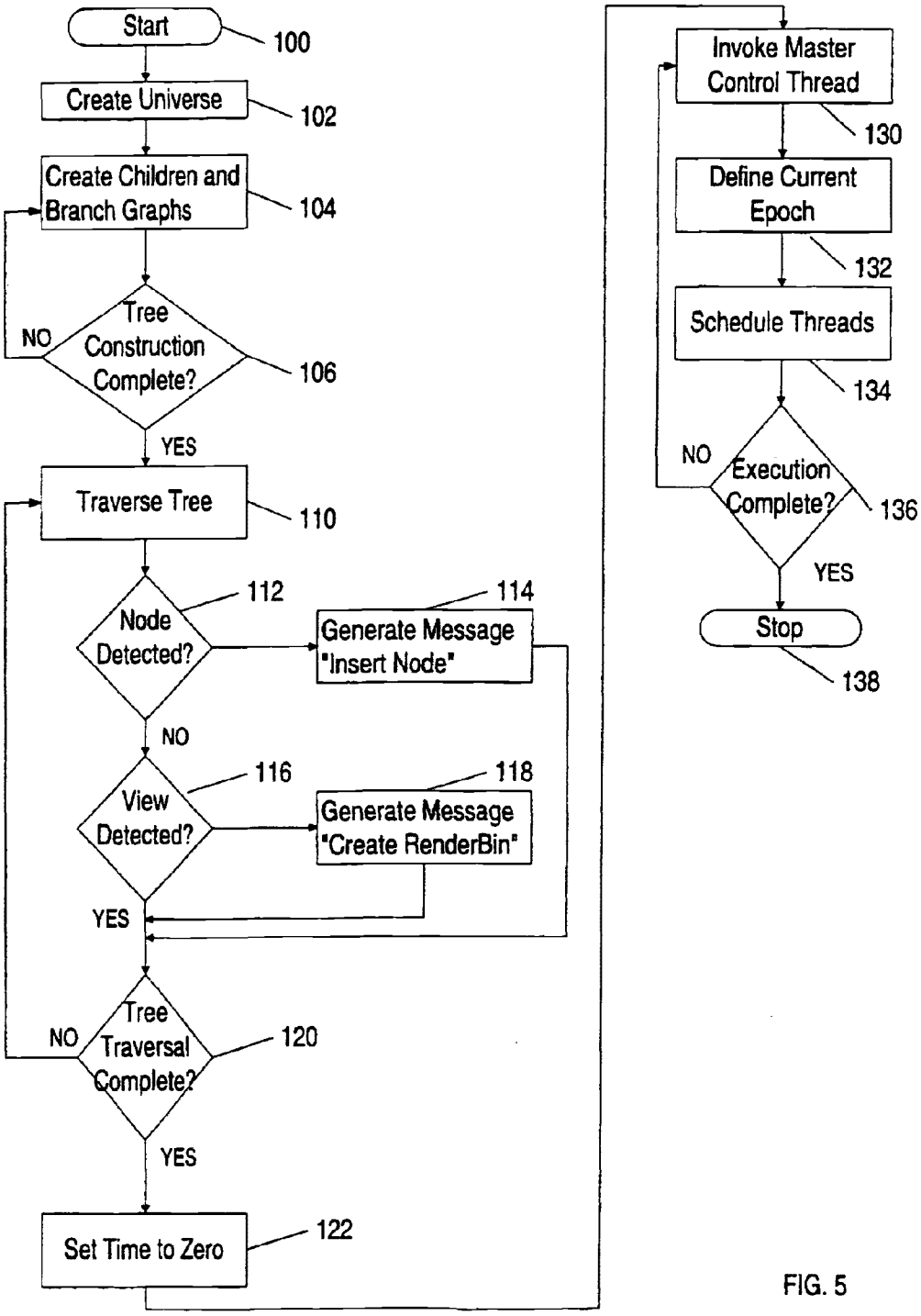


FIG. 5

NVIDIA Ex1002 Page 187 of 1081

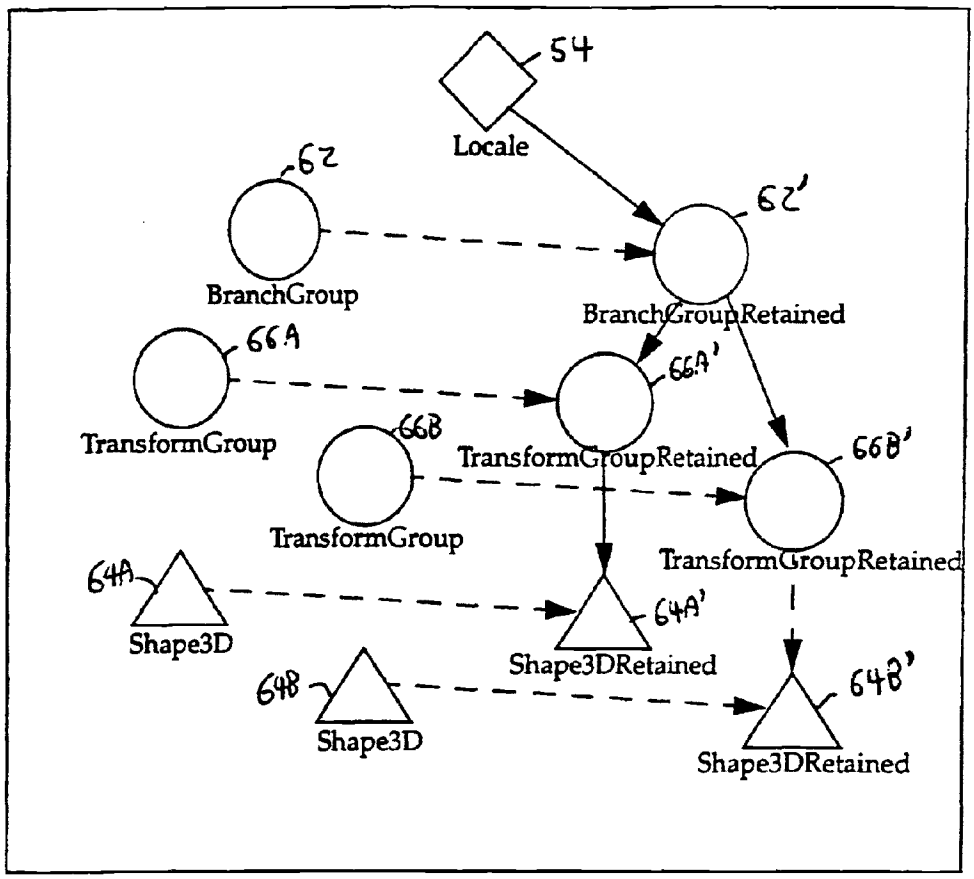


Figure 6 Scene Graph Internal Structure

NVIDIA Ex1002 Page 188 of 1081

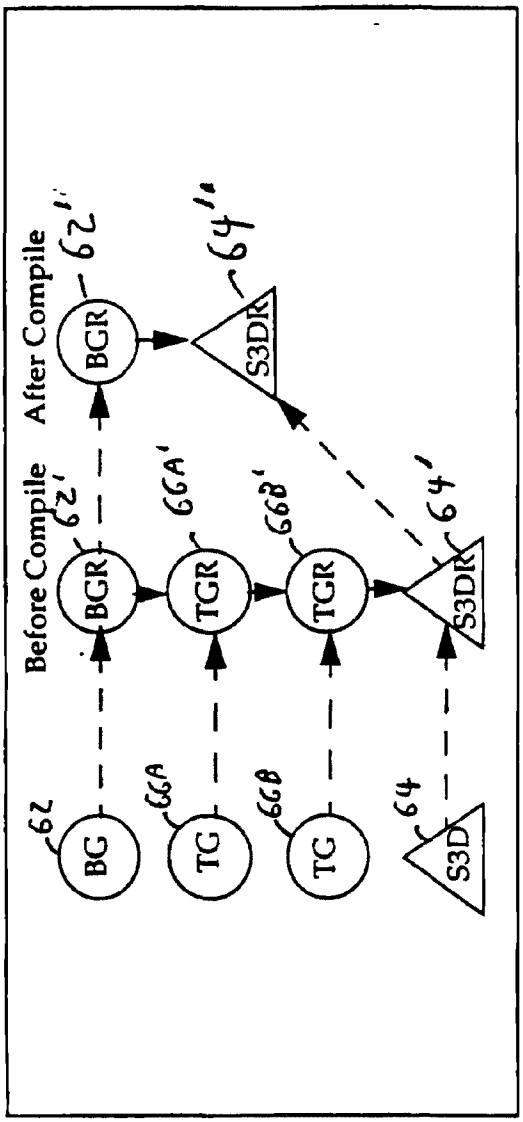
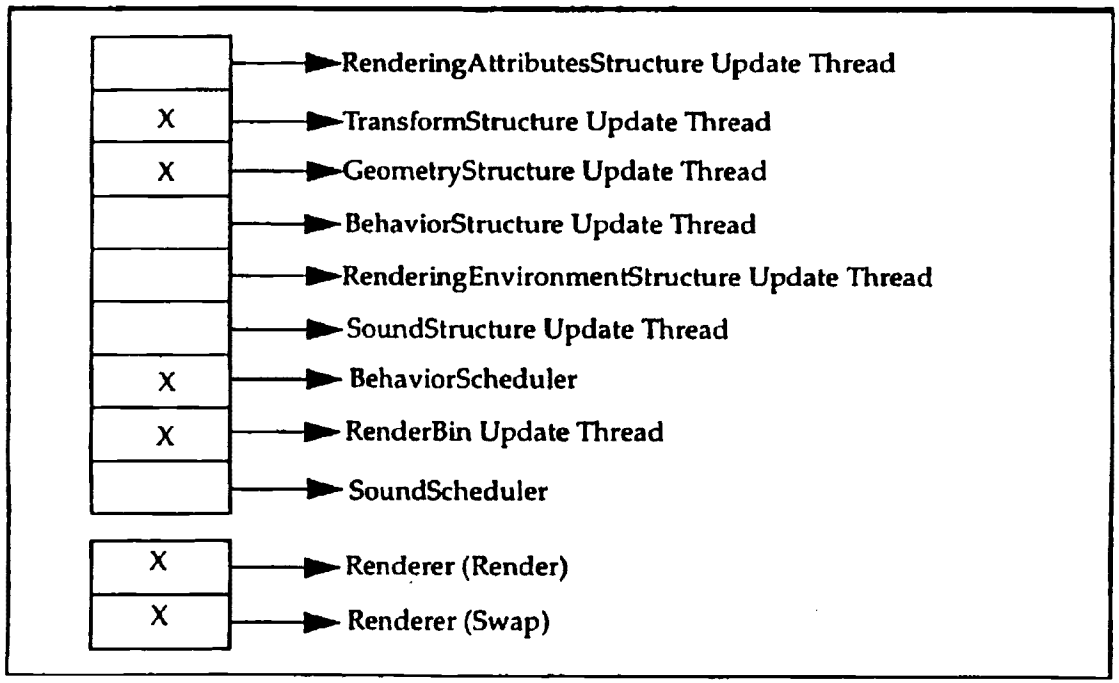


Figure 7 A flattened branch graph

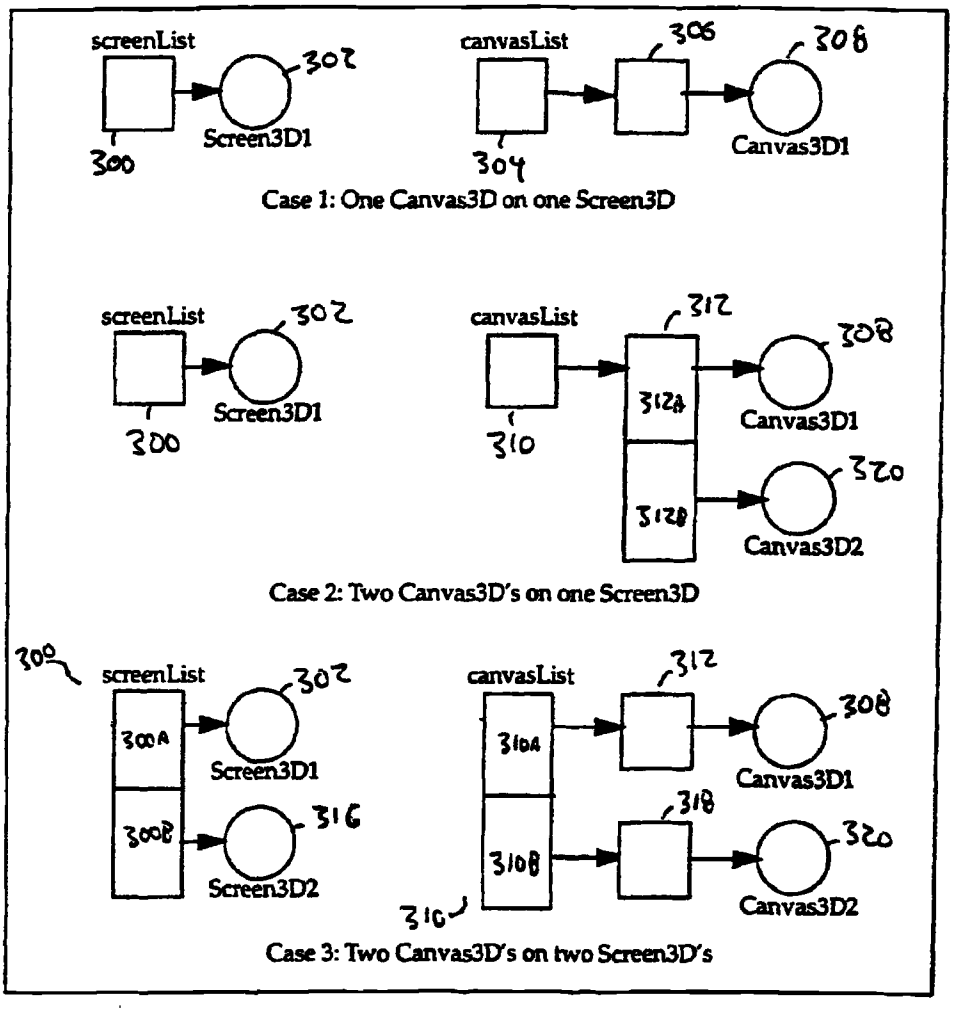
NVIDIA Ex1002 Page 189 of 1081

Figure 8 MasterControl's current work threads



NVIDIA Ex1002 Page 190 of 1081

FIGURE 9



NVIDIA Ex1002 Page 191 of 1081

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 November 2014 (27.11.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/190208 A2

(51) International Patent Classification:
G06F 19/00 (2011.01)

(21) International Application Number:
PCT/US2014/039239

(22) International Filing Date:
22 May 2014 (22.05.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/826,387 22 May 2013 (22.05.2013) US

(71) Applicants: **NEURALA, INC.** [US/US]; The Photonics Center, 8 St. Mary's Street, Suite 613, Boston, Massachusetts 02215 (US). **TRUSTEES OF BOSTON UNIVERSITY** [US/US]; One Silber Way, Boston, Massachusetts 02215 (US).

(72) Inventors: **GORSHECHNIKOV, Anatoly**; 15 Jefferson Street, Newton, Massachusetts 02458 (US). **VERSACE, Massimiliano**; 846 East 3rd Street, Boston, Massachusetts 02127 (US). **BARNES, Tim**; 1305 E. Howell Street, Apt. 101, Seattle, Washington 98122 (US).

(74) Agents: **TEJA, Joseph** et al.; Cooley LLP, 1299 Pennsylvania Ave., Suite 700, Washington, District of Columbia 20004 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) Title: METHODS AND APPARATUS FOR EARLY SENSORY INTEGRATION AND ROBUST ACQUISITION OF REAL WORLD KNOWLEDGE

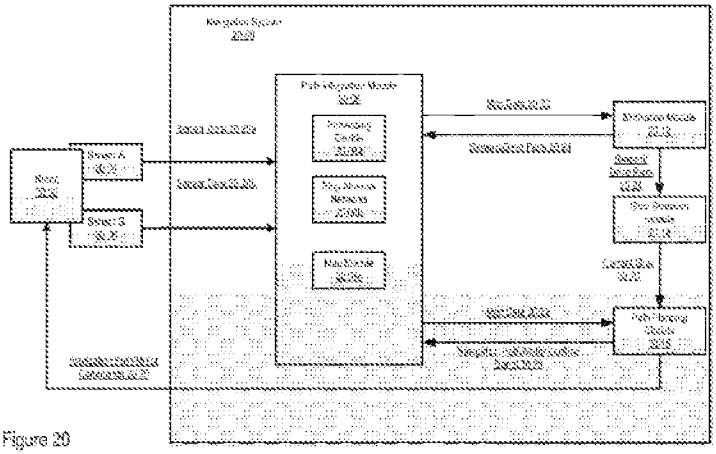


Figure 20

(57) Abstract: The systems and methods disclosed herein include a path integration system that calculates optic flow, infers angular velocity from the flow field, and incorporates this velocity estimate into heading calculations. The resulting system fuses heading estimates from accelerometers, 5 gyroscopes, engine torques, and optic flow to determine self-localization. The system also includes a motivational system that implements a reward drive, both positive and negative, into the system. In some implementations, the drives can include: a) a curiosity drive that encourages exploration of new areas, b) a resource drive that attracts the agent towards the recharging base when the battery is low, and c) a mineral reward drive that attracts the agent 10 towards previously explored scientific targets.

WO 2014/190208 A2

Methods and Apparatus for Early Sensory Integration and Robust Acquisition of Real World Knowledge

CROSS-REFERENCE TO RELATED PATENT APPLICATION

[0001] This application claims priority, under 35 U.S.C. § 119(e), from U.S. Application No. 61/826,387, filed May 22, 2013, and entitled "METHODS AND APPARATUS FOR PATH INTEGRATION AND NEURAL ACQUISITION OF LOCATION ESTIMATE," which application is hereby incorporated herein by reference in its entirety.

GOVERNMENT SUPPORT

[0002] This invention was made with government support under Contract No. NNX12CG32P awarded by NASA. The government has certain rights in the invention.

BACKGROUND

[0003] To date there has been little success in development of robotic intelligent behaviors without reliance on resource-intensive active sensors such as but not limited to laser range finders, sonars, etc. Furthermore, these sensors are often tailored and used exclusively for individual tasks. At the same time, robots do have other multi- purpose passive sensors, but the precision of knowledge based on these sensors leaves much to be desired. The lack of reliable cross-task sensory system is partially due to sensory noise and partially to an approach of independent, stove-piped processing of individual sensory streams.

SUMMARY

[0004] The system and methods described herein is based on a combination of an arbitrary number of sensory inputs correcting each other by compensating weaknesses of one sensor with the strengths of other sensors in the process of early fusion, then processing these sensory inputs through a redundant and robust neural-like system. An exemplary system may be used for a variety of applications, including but not limited to spatial navigation, visual object segmentation and recognition, or robotic attention.

[0005] In some implementations, the system may generate a noise-tolerant distributed data representation model vis generating a set of data cells collectively representing a sensory

NVIDIA Ex1002 Page 193 of 1081

input data point in the range of measurement for a particular type of data, and may use the data to define a spatial or temporal resolution of the set of data cells (e.g., a data representation model) from the data range of the incoming sensory information and the number of cells in the set. A spatial or temporal scale for the representation may be determined using the range of the incoming sensory information, and may be used to generate a set of graph connections among the set of data cells with a first set of weights (in some implementations, these connections are designed to ensure continuity between representation of points in continuous space and some distance measure so that data cell representation reflects the distance between input data points).

[0006] It should be appreciated that all combinations of the foregoing concepts and additional concepts discussed in greater detail below (provided such concepts are not mutually inconsistent) are contemplated as being part of the inventive subject matter disclosed herein. In particular, all combinations of claimed subject matter appearing at the end of this disclosure are contemplated as being part of the inventive subject matter disclosed herein. It should also be appreciated that terminology explicitly employed herein that also may appear in any disclosure incorporated by reference should be accorded a meaning most consistent with the particular concepts disclosed herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The skilled artisan will understand that the drawings primarily are for illustrative purposes and are not intended to limit the scope of the inventive subject matter described herein. The drawings are not necessarily to scale; in some instances, various aspects of the inventive subject matter disclosed herein may be shown exaggerated or enlarged in the drawings to facilitate an understanding of different features. In the drawings, like reference characters generally refer to like features (e.g., functionally similar and/or structurally similar elements).

[0008] Figure 1A shows experimental recordings from a visual cortex that illustrate continuity of sensory data and its representation in the brain.

[0009] Figure 1B shows an example tuning curve for the experimental data shown in Figure 1A.

[0010] Figure 1C shows an example of neuronal representation with narrow on-center (local excitation) and wide off-surround (global inhibition, this is the extreme case when the surround includes all other cells).

[0011] Figure 1D shows an example of a neuronal representation of an animal's heading.

[0012] Figure 2 is an illustration of beats of low frequency created by overlapping two higher frequencies. This can be used in space or time to encode intervals in wide ranges using only a limited set of base frequencies. Vernier scale and hyper-acuity is related to the inverse application of this principle.

[0013] Figure 3 illustrates the process of segmenting objects from their "background", by adaptively combining in a feedforward/feedback process various sources of potentially imprecise/noisy visual and non-visual information to estimate what is figure and what is ground, and create a form-fitting (or object-fitting) mask to enable segmentation of the object of interest.

[0014] Figure 4 depicts a biological path integration system. On the right, Figure 4 names the mammalian brain areas subserving path integration. On the left it describes core modeling work that was combined to create an example PInNAcLE implementation.

[0015] Figure 5 shows the results of PInNAcLE system. Individual motor, vestibular and optic flow (visual) based estimates are plotted without feedback correction described below and show consistent overshoot for motor and undershoot for optic flow. Combined estimate follows the true heading (2D and 3D headings) much closer.

[0016] Figure 6 shows an example flow chart for IMU on the fly calibration algorithm using flow field information.

[0017] Figure 7 shows obstacle map generated by the navigation model presented herein and how it affects path planning search algorithm.

[0018] Figure 8 shows assembly diagram of robotic sensors used in the example embodiment

[0019] Figure 9 shows software components of example robotic embodiment.

[0020] Figure 10 shows an experimental paradigm used to test the navigation model presented herein.

[0021] Figure 11 shows an example trajectory of a simulated robot in a virtual environment.

[0022] Figure 12 shows a map view of a trajectory from Figure 11 as it builds incrementally through time.

NVIDIA Ex1002 Page 195 of 1081

[0023] Figure 13 plots a battery level during the exploration experiment.

[0024] Figure 14 shows results of a model test in a robotic setup described herein.

[0025] Figure 15 provides an overview of processing streams in a mammalian visual system.

[0026] Figure 16 illustrates the result of OpenEye model recognizing rock types in a virtual environment

[0027] Figure 17 depicts Virt-U interface between neural computation and 3D virtual engine.

[0028] Figure 18 provides an example structure of Virt-U classes.

[0029] Figure 19 shows a structure of a proxy layer designed to interface Virt-U with Cog Ex Machina.

[0030] Figure 20 shows an exemplary navigation system architecture.

[0031] Figure 21 shows a flow chart illustrating an exemplary embodiment of path navigation.

DETAILED DESCRIPTION

[0032] This disclosure is subdivided in several sections. The first section describes the basic principles of the biological sensory processing and claims how they can be applied to improve robotic and other engineering applications. The next section describes an example robotic navigational system that is based at least on the principle of the first section and is used to test the applicability of the presented approach. The disclosure presents an example of an artificial visual system that illustrates applicability of the principles in the first section to a completely different domain thus implying the generality of these principles. It also discloses a software layer that allows communication between neural models based on the principles of the first section and built in various software packages with robotic or virtual sensory motor scenarios that allow testing and further development of the above principles.

[0033] Biological Sensory Processing and Engineering Practices

[0034] Sensory information processing in an animal nervous system model differs from conventional engineering approaches in several respects: data representation, horizontal interactions between processing streams within and across modalities, and feedback from higher level results to early processing stages within and across processing streams. Neural-

NVIDIA Ex1002 Page 196 of 1081

like data representation, as well as neural horizontal and feedback interactions deliver advantages in engineering of artificial information processing systems, as detailed below.

[0035] *Data Representation*

[0036] In order to save resources, engineers try to minimize the representation of the sensory data. Whether it is luminance of a visual or radar pixel, acceleration, or sound pitch, conventional systems tend to represent it as a single floating point number with possible duplication or even triplication for the systems where robustness is important. This does not prevent the data from experiencing abrupt changes due to noise. Consequences of this include disregarding a temporal continuity of the physical world around us and (without additional processing) not establishing a relationship between nearby values, thus disregarding a spatial continuity of the world.

[0037] Figures 1A–1D illustrate a neural representation which uses ensembles of cells to represent every data point, which sacrifices the storage space for the sake of robustness and preserving the continuities of the world. Specifically, Figure 1A illustrates this principle for a cell population that codes the orientation of a bar in a visual field of an animal from experimental recordings. In visual cortex a set of cells 100 that represent orientations exist for each point in the visual space, and if a bar of a certain orientation 102 appears at this point not only the cell with the corresponding orientation becomes active 104 but so do cells with nearby orientations 106, the closer their preferred orientations are to the one presented, the higher is the cell activation (so called tuning curve of the cell). As a result, for each data point there is a distributed bell-shaped representation 108 (shown in Figure 1B) that is centered at the precise value of the preferred orientation and gradually decaying as the nearby cells tuning moves away from this value. This distributed representation is ubiquitous in the nervous system, and is used to represent headings in parasubiculum, places in the hippocampus, tones in auditory cortex, and many other sensory readings or processed data. Figure 1C illustrates how the neurons may be connected to achieve such a representation using local excitation 110 and global or long range inhibition 112. Figure 1D shows the activities of the resulting cells showing bell-shaped profile for a particular input (114).

[0038] This representation bears several advantages. First, it is more resistant to damage to the processing hardware. For example, the head direction cells that represent the animal's heading may have tuning curves about 90 degrees wide, which suggests that losing the heading signal completely involves losing at least a quarter of head direction cells. Even then, losing a quarter of the heading cells may affect only a quarter of possible headings.

NVIDIA Ex1002 Page 197 of 1081

[0039] The second advantage of the neuronal representation is preservation of the continuity of the world. The cells in the representation are not standalone memory elements, but a connected ensemble. Usually this connectivity follows some version of on-center off-surround (e.g., Mexican hat) profile, where the cells excite nearby (in terms of data values or tuning curves) cells and inhibit cells that are further away. This connectivity creates an attractor dynamics where it is easy to shift from the current data value to nearby values, but where it is much harder to shift to further away values. This reflects a natural continuity of data: if a person is facing East at one moment, it is virtually impossible for that person to face West in the next moment, and much more likely that the person's heading would still be very close to East the next time the person's heading was requested. From an engineering standpoint, this implementation may also work as a low-pass filter, which can be a first cascade in the sensory processing that serves to reduce or eliminate the high frequency jitter most sensors produce. Therefore, for the fair comparison of storage and processing requirements of state of the art engineering system and biological system we should compare the initial data representation of the latter with a combined data representation and low-pass filter of the former. Using attractor on-center off-surround dynamics may increase the robustness of the representation for continuous values.

[0040] Furthermore, biological systems often utilize multiple scales of on-center off-surround connectivity (corresponding to low-pass filter frequencies) through different lengths of on-center and off-surround connections, resulting in multiple scales of sensory representation. This allows the brain not only to process information at an appropriate scale (e.g., one does not need to know the distance between Boston and New York in feet in order to plan a trip) but also to dynamically switch between scales (e.g., gradually increase one's resolution as one approaches one's destination in New York). Due to a principle of beats phenomenon that occur when one combines two nearby frequencies 202 and 204 and produces a signal with a significantly lower frequency 206 (e.g., as shown in Figure 2), a limited set of base scales can allow for representing a wide range of data (e.g., up to least common multiple of the scales) with high precision (e.g., at least as high as the smallest scale) without a necessity to involve large numbers of cells in the representation. Furthermore, interactions between multiple scales can lead to phenomenon of hyper-acuity, when the information derived from the representation has higher resolution than the original sensory data (this is based on the same principle as Vernier scale used in engineering for

NVIDIA Ex1002 Page 198 of 1081

precise measurements). This principle can be generalized, and can be applicable as long as there is either spatial or temporal extent in the input signal.

[0041] To summarize, using neuronal-like data representation on every processing stage allows robustness of the system, built-in low pass filtering, increase in data precision beyond the precision of a sensor, preservation of spatial and temporal continuity of the signal, and easily scalable ability to handle wide ranges of data with high precision. Bell shaped and often Gaussian profiles of neuronal activity are also very convenient for combining signals from different inputs as discussed in the next section.

[0042] *Horizontal interactions between processing streams within and across modalities*

[0043] A conventional solution for processing multiple sensory streams is to stove-pipe several algorithms, each being finely tuned for a specific task, and combine their outputs for further analysis and guidance of actions. For example, visually estimated self motion is often used to correct odometry-based motion estimates, but it is usually done in discrete steps, rather than continuously, and by resetting the odometry data, rather than fusion of two sensory streams. Each individual algorithm is a result of a long effort to achieve maximal performance for a certain task; however these algorithms in aggregate rarely allow meaningful use of the results of intermediate processing stages. Furthermore, combining the algorithms involves reducing the interdependencies between processing streams to keep the system tractable, leading to little information about how the processing streams relate to each other.

[0044] An approach based on biological principles may involve improving processing, within each stream, using the early and intermediate results of the stream along with other streams. Such an approach may be used, for example, with image segmentation by the visual system. The visual system may be configured to allow a user to build an accurate representation of the world, including relative positions of all objects and how they move with respect to an entity (e.g., a robot). A conventional approach would be to select objects of interest, fine tune detection algorithms for these objects, and use outputs of these algorithms to feed motion detection. A human brain's approach is quite different; it extracts the information from the visual input in parallel, and uses different feature sets 300 of this information to help processing of other features 302 as early as it is useful and long before the objects are detected and identified 304 (e.g., see Figure 3). In the example implementation of an OpenEye system discussed below, determining the boundary of an object can be done using a change in luminance, color, texture, or other surface property, as well as depth

NVIDIA Ex1002 Page 199 of 1081

information that can come from binocular disparity or from accommodation of the eye, and discontinuities in the motion signals like optic flow. All of these cues can be combined on the early stages of processing.

[0045] Furthermore, at this stage the system does not need to know what the objects are, and how many are there; it may be completely agnostic towards identifying objects. As a result, the output of these interactions – a video stream segmented into areas that have similar properties and move together - can be used for a plurality of types of object detection and recognition, thus unifying a significant chunk of processing and making it scalable for any arbitrary number of objects. Additionally, the object-aware part of the system may become independent of the number of sensors used to produce visual segmentation, and can be reused, for example, on different robots with different sensory compositions. Since motion, depth, and different surface properties are processed at different resolution (e.g. motion signals may have coarser resolution than surface property signals), the system may also utilize all advantages that come with multiple scales of neuronal representations outlined in the previous section. Various modalities and even data extraction methods of sensory processing often provide incomplete and imprecise results. Early fusion of the outputs of these algorithms and modalities makes it possible to feed more complete and precise data to computationally expensive high-order processing algorithms.

[0046] Another advantage of early interactions between sensory processing streams may be a shortened response time. For example, for the human brain, interaural differences in sound arrival timings allow the brain to detect the source of sound in space. This detection happens early in the auditory stream in the inferior part of colliculus before the information reaches even the primary auditory cortex. The superior part of colliculus controls eye movements, and a direct link from the auditory stream to visual stream allows humans to move their eyes towards the source of sound long before they can recognize what this sound is. Similarly, a quick estimate of optic flow by a vehicle can be used to determine that there is an object on a collision course with this vehicle and trigger an avoiding maneuver before the nature of the colliding object is processed. The linkage between early sensory processing stages and fast motor responses may reduce response times for important inputs.

[0047] In some implementations (referring to Figure 21), a system may generate a set of data cells 2100, the set of data cells representing sensory input data received from a group of sensors. The sensor may use the set of data cells to both determine a spatial and/or temporal resolution 2102 of the set of data cells, and to determine a spatial and/or temporal scale 2104

NVIDIA Ex1002 Page 200 of 1081

for the set of data cells and the range of available data. In some implementations the resolution may refer to the size of the data set with respect to the range of data. The system may then define a set of weighted connections from the data cells 2106, to the combined representation 2116. Fill in the sensory data 2110. Using the apriori information about each sensor precision, the system may alter the significance of the weights from each set of data points in 2112, e.g., based on the precision of the source of the data represented by each set of weights, and/or the like. The system may weight the sensory input data from each of the sensors 2114 based on the altered sets of weights, and may combine the weighted sensory input data 2116. The weighted sensory data may be fed back into the system in order to update data representation in 2110 as in 2118. Iterate through the loop between 2110 and 2118 as long as new sensory data comes in.

[0048] Another illustration of across modality interactions is a Path Integration and Natural Acquisition of Location Estimate (PInNAcLE) system based on biological path integration illustrated in Figure 4. In some implementations, the PInNAcLE system transforms velocity data 400 recorded from the agent's sensors and actuators into the robot's world position. The basic element of the system is a ring attractor network. Here the inputs from three sensory modalities 402 are combined: motor outflow, which precedes the motion, but can be incorrect since the body might be restrained and the effectors might slip; IMU input, which is slightly delayed and has issues with detecting very slow motions (see Table 1); and optic flow, which is good for slow rotations but becomes increasingly inaccurate as the speed increases and cannot estimate linear motions without additional information.

Size\Normalized input	0.0001	0.0002	0.0004	0.001	0.002	0.004	0.01	0.02	0.04
36								0.14	0.34
72							0.085	0.18	0.36
180		0.002	0.004	0.01	0.018	0.036	0.092	0.184	0.368
360	0.001	0.002	0.004	0.009	0.018	0.037	0.092	0.184	0.368
720	0.001	0.002	0.0035	0.009	0.0185	0.037	0.0925	0.1845	0.3685

Table 1 Calibration of a ring attractor from Figure 1. Left column shows the size of a network in cells, top row shows a normalized between 0 and 1 angular velocity input, values in the table show attractor rotation in degrees per second. Pink shaded cells show where the input was too small for the

NVIDIA Ex1002 Page 201 of 1081

attractor to respond, yellow shading shows where attractor performed at the edge of loosing precision due to small values of the input.

[0049] Each of these streams has advantages and disadvantages relative to others and a proper fusion of this information within the PInNAcLE system allows these streams to capitalize on the advantages of early sensor fusion and also to self-correct problems with different sensors. The resulting system can reliably track the vehicle position during extended periods of exploration.

[0050] PInNAcLE derives estimate of current heading from each of the three inputs and represents these estimates as shown in Figure 1B. The Gaussian shape of these signals simplifies their combination, as the sum of Gaussians that are close enough together (how to ensure this condition is discussed in the next section) is also a Gaussian. But simple sum may not suffice; in some implementations, the motor input may only be considered if it is backed up by either vestibular or optic flow, while the latter two may be weighted proportionally to the movement speed, with optic flow having more weight for slow speeds and vestibular having more weight for fast speeds. The resulting heading estimate 500 may take advantage of predictive power of motor outflow and high precision of vestibular input at high speeds and optic flow at low speeds (e.g., see Figure 5).

[0051] In some implementations, optic flow (e.g., the set of local motions perceived across the visual field at a particular time), provides a distributed and robust cue for navigation. Biological organisms use optic flow for navigation; for example, when human visual and vestibular senses are in conflict, the visual sense of self-motion can often times overpower the vestibular one. An optic flow system is included in the model (described below) in order to leverage this source of information as a complementary signal to vestibular and motor commands for path integration. The custom algorithm used to determine optic flow is a hybrid of two models.

[0052] Firstly, an initial optic flow field is calculated using monogenic filters, which have been useful for fast optic flow computation. These filters are computationally efficient (especially on a parallel GPU-enabled platform) and naturally illumination-invariant because they are based on a local image phase only. Because the signal is local, however, it is often noisy; this noise can be dampened by spatiotemporal smoothing.

[0053] Thus, a method of smoothing the monogenic signal was developed based on a motion integration algorithm. In some implementations the algorithm estimates agent velocity parameters from the optic flow field according to a pinhole camera model.

[0054] *Monogenic filters*

[0055] The optic flow field (e.g., a 2D vector field over an image) is first constructed by calculating a set of features over the image and then correlating those features over time. The optic flow system extracts local image phase information through a set of monogenic filters over a set of spatial scales s. Monogenic filters are based on Gabor wavelet filters, which balance some spatial and some frequency selectivity. They can be parameterized by their center spatial frequencies; the algorithm uses a particular set of center frequencies spaced geometrically according to

$$\mu_s = \mu_0 \cdot \gamma^{-s},$$

where $\mu_0 = 0.25$ is the highest-frequency filter scale, subsequent filters are scaled by a factor $\gamma = 1.3$, the number of spatial scales $n_s = 6$, and so the set of spatial scales $s \in \{0, \cdots, 5\}$.

Monogenic filters are based on bandpass filters $G_{f_x f_y s}$. The general shape of one bandpass filter is given by a Gaussian function in polar coordinates, which produces a ring in image frequency space:

$$F_{f_x f_y}(\mu, \sigma) = \frac{1}{2\pi\sigma_r} \exp\left(\frac{(f_r - \mu)^2}{-2\sigma^2}\right),$$

where the radial component f_r is given by

$$f_r = \sqrt{f_x^2 + f_y^2}.$$

[0056] This ring can be re-parameterized exclusively by the spatial scale s, in which case the width of the ring scales with the center frequency μ_s:

$$G_{f_x f_y s} = F_{f_x f_y}(\mu_s, 0.85\mu_s).$$

[0057] This bandpass filter estimates the amplitude of local frequency at an image; the local angle and phase at that frequency is estimated with filters that select for horizontal or vertical frequency energy. These two real-valued filters can be combined into one complex filter $H_{f_x f_y}$ to save on computational resources:

$$H_{f_x f_y} = \frac{i f_x - f_y}{f_r},$$

NVIDIA Ex1002 Page 203 of 1081

[0058] To get the complete monogenic signal, the image I_{xyt} is transformed into ordinary frequency space with the Fourier transform $\mathcal{F}: (x, y) \rightarrow (f_x, f_y)$, multiplied by the set of bandpass filters, and then transformed back into the original image space.

$$\vec{a}_{xyst} = \begin{bmatrix} \mathcal{F}^{-1}\left[\mathcal{F}(I_{xyt})G_{f_x f_y s}\right] \\ \Re(\mathcal{F}^{-1}\left[\mathcal{F}(I_{xyt})G_{f_x f_y s}H_{f_x f_y}\right]) \\ \Im(\mathcal{F}^{-1}\left[\mathcal{F}(I_{xyt})G_{f_x f_y s}H_{f_x f_y}\right]) \end{bmatrix} = \begin{bmatrix} a^{(1)}_{xyst} \\ a^{(2)}_{xyst} \\ a^{(3)}_{xyst} \end{bmatrix}.$$

[0059] Note that, since I_{xyt} and $G_{f_x f_y s}$ are both real-valued, and the resulting filtered signal is also real-valued ($\vec{a}_{xyst} \in \Re^3$). Changing this monogenic signal into spherical coordinates can recover the three components of local image phase: amplitude, angle, and phase roll.

The difference in this monogenic signal over time produces an optic flow field. The monogenic filter signals are specifically compared between time frames for a change in local estimated phase, which is taken as an estimate of local image movement in the direction of the monogenic signal's estimated phase angle. The following equation, based on quaternion rotation, keeps the movement vector in the same monogenic space:

$$\vec{b}_{xyst} = \begin{bmatrix} \vec{a}_{xys(t-1)} \cdot \vec{a}_{xyst} \\ a^{(1)}_{xys(t-1)}a^{(2)}_{xyst} - a^{(1)}_{xyst}a^{(2)}_{xys(t-1)} \\ a^{(1)}_{xys(t-1)}a^{(3)}_{xyst} - a^{(1)}_{xyst}a^{(3)}_{xys(t-1)} \end{bmatrix}.$$

Spatiotemporal smoothing

[0060] The monogenic optic flow signal is efficient and local, but this spatial locality can produce unwanted noise in the resulting optic flow field. This noise can be reduced through a process of spatiotemporal smoothing. The following method of smoothing the monogenic signal was loosely based on a model of biological motion integration. This motion integration process allows for some amount of local smoothing while retaining the computational efficiency of monogenic filters. More complicated smoothing algorithms may be useful for scenes with independently moving objects.

[0061] The main smoothing process, loosely identified with the primary visual cortex, V1, takes the input flow signal, averages that signal over space and time, and incorporates feedback that averages over spatial scales. The process of averaging over spatial scales

roughly corresponds to the functional role of a downstream motion-sensitive brain area, the middle temporal area (MT). This averaged value can be read by other parts of the model as the final optic flow signal.

[0062] *Egomotion estimation*

[0063] Finally, the optic flow signal is used to estimate the camera's rotation, and thus the rotation velocity of the agent. This is done by projecting the optic flow field onto a template field that linearly scales with camera rotation speed. The template field only depends on the pixel position (x, y) and the camera parameters, namely, the focal length α and the physical pixel size β. Note, however, that the pixel indices range from negative to positive rather than starting at 0 or 1; for example, for a 640 × 480 image, $x \in \{-319.5, -318.5, \ldots, 319.5\}$ and $y \in \{-239.5, -238.5, \ldots, 239.5\}$. The template field is given by

$$\vec{R}_{xyt} = \frac{-1}{\alpha}\begin{bmatrix}\alpha^2 + (\beta x)^2 \\ \beta^2 xy\end{bmatrix},$$

and the projection operation can be written as

$$r_t = \frac{\sum_{xy} \vec{R}^T_{xyt} \cdot \vec{v}_{xyt}}{\sum_{xy} \left\| \vec{R}_{xyt} \right\|^2}.$$

[0064] If a spherical camera model such as the one initially proposed were used instead, then the template field $\vec{R}_{xyt}$ would be independent of pixel position. This simplification would allow for less memory storage but offers little to no speed advantage over the current image plane method. This contrasts with the calculation of linear velocities of optic flow, which depend on an estimate of depth at every pixel that changes over *time* as well as space, requiring more computational resources and communication. Unfortunately, modeling a spherical camera in the virtual environment poses particular challenges because OpenGL is heavily optimized for graphics in Cartesian coordinates. A spherical camera projection would require that all straight lines between triangle vertices to be deformed into curved lines on the camera sensor. The results determined by simulation shown below show that the pinhole camera model is sufficiently fast and accurate for egomotion estimation in the virtual environment.

[0065] Inertial measuring units (IMUs) have three sources of error with decreasing magnitude: Static Bias, Dynamic Bias and Random White Noise. Static bias is reduced by recalibrating the IMU between runs when the vehicle is stationary. Dynamic bias occurs

NVIDIA Ex1002 Page 205 of 1081

during the mission while the vehicle is underway and if not addressed can lead to drift – where the error from the previous position estimate further impacts the next position estimate. The correction of this drift can happen independently for rotational IMUs (gyroscopes) and translational IMUs (accelerometers) from the segregated optic flow or other flow field for example sonar depth field through time. Full flow field can be separated into depth-invariant component that represents rotation of the vehicle and depth-dependent component that represents its translation. Periods of time when there is no rotational flow can be used to recalibrate rotational IMUs, and periods with no translational flow can be used to calibrate translational IMUs. Effectively, this is a natural continuation of the weighted approach described in the previous paragraph: when the perceived motion is zero optic flow or sonar flow has its maximal effect on the position estimate.

[0066] Sensory fusion may weight the importance of incoming signals and modalities depending on the state of the system and the advantages of individual sensory streams. Some states shall be used to perform on-the-fly recalibrations of the sensors that can show significant drift during extended performance.

[0067] Figure 6 presents an example flow chart for this algorithm. System acquires feature field, for example an optic flow field on every iteration or frame 600. If there is no signal the field will be flat with all values equal 0, so the system proceeds to the next frame 602. If there is signal, then the system computes a velocity field by taking difference between this frame and the previous frame 604 and acceleration field by taking the difference between velocity field from the previous frame and the newly computed one 616. It may then determine whether the velocity field is flat 606. The velocity field can be flat only when there is no linear motion 614, so the system can override other inputs and ensure that the coordinate of the agent does not change its representation. After dividing the velocity field by distance 608 the system may determine if the field is flat 610; dividing may produce a flat field only if there is no angular motion 612, so the system can be sure that the heading of the agent does not change. Acceleration field 616 is used in a similar fashion to determine angular and linear accelerations and signal an opportunity to recalibrate IMUs when respective accelerations are 0 (e.g., the system checks to see if the acceleration field is flat 618, and may calibrate the accelerometer 626 if the field is flat; if the field is not flat, the system may divide the acceleration field by distance 620, and if the field has become flat 622, the system may calibrate the accelerometer 624.

NVIDIA Ex1002 Page 206 of 1081

[0068] Linear displacement is first estimated along three axes in the plane; the use of three axes instead of two provides redundancy and error detection (band cells 404 in Figure 4). Each axis has a set of corresponding ring attractors – all having a preferred direction aligned with the axis, but each having different spatial scales: the displacement covered by one full rotation of attractor around the ring. The neural system achieves its best localization results with spatial scales related to each other as different prime numbers (e.g. 7 to 11 to 17 for three scales). The merged heading estimate is filtered through a pair of Gaussian filters (one positive and one negative, centered 180 degrees apart from each other) to produce a cosine of an angle between the current heading and a preferred direction of a linear displacement ring attractor. This cosine, multiplied by the linear velocity of the agent (estimated by different sensory inputs), serves as an input to the corresponding set of ring attractors. The merged displacement estimates along axes are combined together into 2D displacement within the corresponding spatial scale (grid cells 406 in Figure 4). Finally, 2D maps of different spatial scales are merged together to produce a unique (within a map) current location estimate for the robot with respect to its starting location (place cells 408 in Figure 4).

[0069] Note that the final output of PInNAcLE system is an estimate of the agent's position and orientation in space, but instead of estimating these outputs from three different sensory inputs independently the information is combined on the stages of heading estimation and linear displacement estimation, and these combined more precise estimates are used downstream to produce final results. Similar to the visual processing example above, this downstream system also is independent of the sensory composition of early processing in PInNAcLE.

[0070] To summarize, unlike the vertical streams of processing used in conventional approaches, the biological approach suggests high level of horizontal interactions between streams and clear separation of processing stages in terms of defining intermediate outputs that can be used elsewhere in the system.

[0071] *Feedback from late stages to early stages within and across processing streams.*

[0072] Introduction of continuous delayed feedback in the system induces oscillatory regimes that usually complicate both mathematical analysis and performance tuning of the system. As a result, state of the art solutions for sensory processing tend to utilize corrective feedback mainly in the form of discrete resets. Furthermore, as a consequence of vertical stove-piping, the results of many sensory estimates are not trustworthy enough to use as a

NVIDIA Ex1002 Page 207 of 1081

corrective feedback. For example in loop-closure solutions visual input is used to recognize whether the agent completed the loop during navigation, and if so it is used to correct odometry-derived position estimate. Rarely if ever odometry is used to affect decision making on whether the loop was closed. As a result, the system makes errors when a view is identical in different parts of the environment (which happens often indoors), while a feedback from odometry could have eliminated a significant portion of these errors that just could not be loops even with low precision of odometers.

[0073] The brain on the other hand is an inherently oscillating system that shows us how oscillations can be utilized for synchronization, noise reduction, and dynamic control. Since brain puts oscillations to use rather than avoid them, the dynamic feedback is a key mechanism in the brain functionality and can be utilized for much more than just corrective input. Figure 5 shows how in PInNAcLE the fast motor input by itself tend to overshoot the motion, while optic flow based signal tends to undershoot, primarily because it takes longer to process and thus comes with a delay. Results of heading estimates from all sensors are merged together in a weighted fashion described in the previous section, allowing the system to take advantage of the best ranges of precision of individual sensors. The result of this merge is computed in the ring representing the final estimate and then provided back to individual sensors, correcting the individual errors and ensuring that all sensory estimates stay tuned to each other and do not diverge with time. In some cases, early corrective feedback to sensory readings reduce error in the system on early stages of processing.

[0074] Providing corrective feedback to these signals from a combined version on one hand introduces more inertia in the system (consider it another low-pass filter) and on the other hand allows keeping the Gaussians of all individual estimates within a short range of each other (as discussed in the previous section it is required for successful combining of these signals). Furthermore, the level of feedback can determine how much individual channel matters: increasing feedback to optic flow based estimate will reduce the lag of this estimate, but also make it less contributing input. Dynamic weights in the merging and feedback allow manipulation of the importance of each individual sensory stream for the final result during the operation of the system and depending on the conditions changing due to external events as well as the robot's own actions.

[0075] Since the fusion of different inputs happens in the system as early as possible, it reduces the error at early processing stages and prevents error propagation through later stages of processing. The resulting system a) produces positional estimates that are as good as

NVIDIA Ex1002 Page 208 of 1081

the best precision sensors it employs allow it make, b) can merge an arbitrary number of these sensory streams, and c) will continue to operate as long as at least one of these sensors is operational. Early fusion of sensory streams allows the rest of the system to be agnostic to the number and quality of sensors, which makes the system more robust and maintainable.

[0076] Correction of drift in the sensory information is not the only, and not even the main reason to use feedback in sensory processing. The main advantage the feedback can provide is the speed of processing. In the example implementation of active vision system OpenEye discussed below a visual object recognition cascade of the system may recognize a blob in a bottom left part of the visual field as a pickup truck. Motion detection sensory system may detect that it is moving to the right. The motor outflow system has full information about how the agent plans to move the camera. Given that all these processes are continuous, it automatically entails that in the next moment of time the OpenEye system knows precisely where in the visual field this truck shall be by combining feedback from object recognition with low level sensory/motor information.

[0077] This may result in the ability of OpenEye system to avoid further engagement of all the complicated cascades of the system, all the system has to do is to feed back to the early stages the expectation of what should be there and check how well it matches the input. If there is a match, then the expensive object recognition processing is not necessarily needed. If there is a mismatch, then the system engages full power to process why its prediction was wrong and how to correct it. This ability to disengage 90% of the computational capability and only concentrate on important parts of the input may outweigh the initial complexity the system. This advantage in terms of the cost of operating resources may manifest itself in wide spread applications as the energy efficiency becomes more and more important. Feedback across different modalities creates the predictive power in the intelligent system, so it can work in a predictor-corrector mode. This mode frees computational resources when correction is not needed.

[0078] The process described in the previous paragraph can be taken one step further by creating a memory map of a virtual visual field not limited by the current field of view. An example OpenEye implementation can place certain objects in the virtual field of view, turn 180 degrees, then turn back to the original view and expect to see the previously recognized objects there.

[0079] **Robotic Navigational Systems**

[0080] As an illustrative example of principles discussed in the previous section, this section provides a description of a robotic navigation system based on these principles.

[0081] Development of a biologically-inspired, robust, low-power, multi-component neural system able to perform self-localization and mapping will enable robots to autonomously navigate novel terrains without a global navigation satellite system (GNSS). By including the ability to learn about an environment as it explores, robotic agents will be able to autonomously negotiate novel terrains and send relevant, intelligently preprocessed information back to a human controller.

[0082] In some implementations, the system and methods described herein reduce development costs and time by allowing robots to be trained rather than explicitly programmed. The systems and method described herein can be implemented into robots, autonomous vehicles, and other moving agents for use in a plurality of applications, including: defense (e.g., Unmanned Aerial Systems (UAS) surveillance, patrol, rescue, demining); business (e.g., telepresence); home (e.g., cleaning, lawn mowing, toys); healthcare (e.g., remote diagnosis, assistive living); and agriculture (e.g., pot moving, autonomous seeding, crop assessment, wildlife conservation).

[0083] In some implementations, the navigation system and method described herein include neural systems mimicking the ability of biological organisms to navigate their environment as well as build usable and correctable representations of these environments through learning without a GNSS and active sensors.

[0084] The navigation system can navigate, map, and detect objects in land and aerial environments. Further capabilities of the system include visual learning and search, collision avoidance, a human-robot interface for robot training and deployment, and low-power hardware implementation of these algorithms.

[0085] According to one aspect of the disclosure, the system includes a path integration system that calculates optic flow, infers angular velocity from the flow field, and incorporates this velocity estimate into heading calculations. The resulting system fuses heading estimates from accelerometers, gyroscopes, engine torques, and optic flow to determine self-localization. The system also includes a motivational system that implements a reward drive, both positive and negative, into the system. In some implementations, the drives can include: a) a curiosity drive that encourages exploration of new areas, b) a resource drive that attracts the agent towards the recharging base when the battery is low, and c) a mineral reward drive that attracts the agent towards previously explored scientific targets.

[0086] In some embodiments, the system's biologically inspired navigational system includes a number of subsystems for an autonomous navigation. The subsystems can include: Path integration system for maintaining a sense of current position and direction relative to starting position and direction. Spatial representation incrementally built as the agent explores its environment. Path planning system that uses information from the spatial representation and path integration system to explore, return to base, revisit objects of interest, or avoid obstacles. Drive selection system to balance battery recharging, exploration, and return visits to explored locations of interest. Sensory system used to provide vestibular, motor, and optic flow information to drive and correct the path integration system.

[0087] The present disclosure also provides experimental results of the disclosed system in both a virtual environment and as implemented into an autonomous rover.

[0088] The navigation system can include an artificial brain model that constitutes integrated perceptual, decision-making, and motor systems. Using the navigation system, the agent can interact with its environment in an intelligent way based on its internally generated goals. The model can contain many interacting subsystems. For example, the subsystems can include a Path Integration and Neural ACquisition of Location Estimate (PInNAcLE) system. The PInNAcLE system can transform velocity data recorded from the agent's sensors and/or actuators into a world position of the agent. This world position can be used by the other modules described herein to learn positions of objects in the world. The PInNAcLE system is described in below. Another subsystem can be the MOtivation, Reward, and goal SELection (MoRSel). In some implementations, the MoRSel subsystem codes agent drives, learns maps where each drive can be satisfied through corresponding rewards, and/or selects a goal map location based on relative drive strengths. The MoRSel system is described below. Yet another subsystem can include the Spatial Planning and Allocentric Representation of Knowledge (SPARK) system. The SPARK subsystem can plan a path and provide a goal heading direction based on the current location, goal location, and/or the locations of learned obstacles. The SPARK system overview is given below.

[0089] *Path INtegration and Neural ACquisition of Location Estimate (PInNAcLE)*

[0090] The navigation system's PInNAcLE subsystem follow the biological path integration system outlined in Figure 4 and can be based on joint information from a number of sources. The PInNAcLE system can include input from motor commands producing self-motion of the agent, input from vestibular or inertial sensors (e.g., accelerometers and gyroscopes), and optic or other sensory flow fields (e.g., from video cameras or sonar

NVIDIA Ex1002 Page 211 of 1081

sensors). The PInNAcLE system can maintain the agent's sense of location within the environment. The motor output commands can add the ability to anticipate movement before the agent performs the movement, and optic flow can add precision to the estimates of angular velocity especially important for low-velocity rotations as described in above. Detailed description of the optic flow computation used to generate result presented here is provided below.

[0091] n some implementations (e.g., Figure 20), a navigation system 2000 may be able to connect to a robot 2002 in order to provide guidance 2030 on where to travel within an environment. For example, the robot may have at least sensors A 2004 and B 2006, which each may correspond to a plurality of types of sensor data 2020a and 2020b. In some implementations, sensor types may include IMU sensors, motor outflow sensors, and optic flow sensors.

[0092] The robot may send the sensor data to the path integration module 2008, which may include a processing device 2008a (e.g., a processor and/or other computing device), ring attractor networks 2008b, and a map module 2008c. In some implementations there may be four ring attractor networks for IMU sensors, four ring attractor networks for optic flow sensors, and four ring attractor networks for motor outflow sensors. In some implementations there may be one ring attractor network per sensor for determining a heading, and one ring attractor network per sensor per spatial scale for linear displacement. The processing device may be able to process the sensor data from the sensors by weighting the sensor data (e.g., in relation to other sensor data) processed by the corresponding ring attractor network, and by combining the sensor data. The combined data may then be used to generate environment data (e.g. information about the robot's current position in its environment and/or other data related to the environment). The map module may use the environment data to generate a plurality of maps related to different data (e.g., different maps may be generated for different goals, different types of sensor data, and/or the like. In some implementations, the map module may be separate from the path integration system.

[0093] The path integration module may interact with a motivation module 2012, a goal selection module 2014, and a path planning module 2016 in order to generate those goals and/or to generate other data. For example, the path integration module may provide the map data 2022 to the motivation module so that the motivation module can determine reward/drive pairs 2024 to provide to the goal selection module for the robot. The reward/drive pairs may be determined by examining maps and determine potential reward

NVIDIA Ex1002 Page 212 of 1081

items in the environment and codes and/or values to relate to those reward items. The goal selection module then may weigh the reward/drive pairs against one another to determine at least one pair to prioritize. The prioritized pair may be become a goal 2026 which may be sent to the path planning module. The path integration module may send the map data to the path planning module, in order to use the map data and the system's spatial understanding of the environment, and to use the goal criteria in relation to the environment, in order to generate a path 2028 from the current position of the robot to the robot's goal. The path planning module may forward this path to the robot in order to allow the robot to understand where to travel next 2030 and to path integration module as motor outflow signal to be integrated as expected path 2028.

[0094] Both path planning and path integration modules can utilize the same internal spatial representation system that follows the principles outlined in above and holds several specific 2D maps in it. A human readable map can be constructed by blending the goal selection maps activated by each drive (drives can be color coded to mark the corresponding reward type) and an obstacle map. The resolution of the map can depend on the spatial scales of grid cells in the PInNAcLE subsystem and can be adjusted according to the user's requirements.

[0095] *MOtivation, Reward, and goal SELection (MoRSel)*

[0096] The MoRSel (Motivation, Reward, and goal SELection) subsystem is an extensible internal motivation framework that can allow for the definition of an arbitrary variety of agent rewards. Each type of reward can be matched with a corresponding drive that, in some implementations, can be calculated as a function of the agent's internal state. These drives can be analog values that may be modulated or explicitly set by an end-user. When the agent navigates to a location that contains a combination of rewards, it can learn the global position of those rewards by increasing the value of synaptic weights between the corresponding drives and the place cell that codes for the agent's current location. Mapping reward values to drive weights may allow the agent, upon the growth a particular drive, to reconstruct an allocentric reward map in order to return to a reward location even after it has traveled away from that position.

[0097] The MoRSel subsystem can contain reward-drive pairs. In some implementations, the subsystem contains three reward-drive pairs, where each drive or reward signal can range from 0 to 1. The MoRSel subsystem can switch between navigation goals based on the value of drives and learned potential rewards. Exemplary reward-drive pairs can include: 1. A

NVIDIA Ex1002 Page 213 of 1081

resource drive that monitors the internal battery level and/or similar depletable resources. For example, as the battery drains, the drive to find a source of replenishment increases. 2. A mineral drive that is a drive to a target of interest. For example, the mineral drive can be satisfied when the agent is in proximity to a target rock that has a reward value. In some implementations, the reward value is designating as a scientific or other interest attached to it for example rocks on Martian surface that can potentially contain water. 3. A curiosity drive that can be a drive to explore an environment. In some implementations, there may be no explicit rewards that have been placed in the environment, but the curiosity drive is instead remembers all previously visited positions in its internal environment map and drives the agent to unexplored locations. This drive can decay once the agent's environment has been fully explored.

[0098] The navigation system can learn the positions of rewards and obstacles by navigating to those locations and making memories that record the presence or absence of motivationally (i.e., reward / deterrent) stimuli. These memories may be stored as synaptic weights between the drives described above, and global position coded by the Path Integration (PInNAcLE) system. These weights can be updated for example using simple associative learning rules, temporal difference reinforcement rules or Q-learning method.

[0099] In some implementations, the navigation system can treat obstacles and rewards differently (i.e., each has a separate map and/or a different learning rule). A separate map may be used because obstacles restrict path planning through the inhibition of the bidirectional search described below. Obstacle weights can decrease when the agent is located at that position, but the weight can increase when the agent is at a nearby location and simultaneously sensing an object in the direction that would lead to the weight's coded location. The process of transforming a local proximity or touch signal into a world position is described further below.

[0100] The goal selection system may balance the strengths of one or more drives with the agent's confidence of the availability of rewards at different environment locations in order to choose where to navigate to substantially satisfy its currently active drives. Before selecting a location as a goal, the system may mask locations that would either place the agent at the position of an obstacle or out of the range of positions that the agent can represent. . In some embodiments the presence of an obstacle attracts the mask value towards 0, and may keep it from being selectable as a goal location. In some embodiments, the mask value slowly recovers if the obstacle weight shrinks due to evidence that the location is

navigable. In order to keep the agent from leaving the area representable by the PInNAcLE system, the positional areas of the mask may be silenced so that they may not be selected as potential goal locations.

[0101] After the candidates for the goal are selected, a new goal position may be chosen through a competitive process, for example a winner-take-all function. Once this winner is selected it is held in memory either until a reset signal occurs or until the location is reached by the agent. If the reset signal happens before the goal is reached, the whole process of goal selection repeats and a new goal (potentially different from the old one) is selected.

[0102] **Spatial Planning and Allocentric Representation of Knowledge**

[0103] The Spatial Planning and Allocentric Representation of Knowledge (SPARK) system may translate agent senses and actions between egocentric and allocentric coordinates in order to control agent exploratory behavior. On one hand, SPARK transforms the agent's current allocentric position from the PInNAcLE system (above) and the goal location selected by MoRSel into a substantially optimal heading for quickly and successfully reaching the goal location and also driving the agent motors. On the other hand, the touch and proximity senses provide local egocentric information as to the proximal location of obstacles, which SPARK transforms into an allocentric spatial position to be learned as an obstacle position by MoRSel.

[0104] *Allocentric representation of knowledge*

[0105] The allocentric spatial representation in the system may model neural representations found in the mammalian brain. Place cells produced by the PInNAcLE system (above) may provide a 2D Cartesian coordinate of the agent on the surface, and head direction cells provide the heading. Both systems may use similar representations as a bump of activity in the neural field that is centered on a corresponding position or direction as discussed in above; the difference being that the positional map is 2D and the directional map is wrapped-around in 1D. The positional signal is may be used for goal and obstacle mappings, and the directional signal may be used in coordinate transformations between egocentric and allocentric representation for obstacle mapping and motor command calculation. Both systems may maintain their relative position and heading with respect to initial position and heading; anchoring these systems to absolute coordinates.

[0106] When the agent registers contact with an obstacle, SPARK translates this information into an allocentric world location of the obstacle that is learned as un-navigable when planning future routes through the environment. The obstacle's heading, compared to

the agent, is rotated according to the agent's internal heading in the PInNAcLE system to give an allocentric direction of the obstacle relative to the agent. The position of the agent, represented by an active place cell in the agent's PInNAcLE system, may be used to increase the obstacle weight for an adjacent place cell that represents the position where the obstacle lies according to the allocentric direction calculated above. In this way the proximal obstacle may be learned on an allocentric obstacle map.

[0107] A chain of neural populations may also do the inverse conversion of allocentric to egocentric coordinates in order to provide motor commands. An example system may convert the desired destination coordinate into the rotational velocity motor command, which then controls navigation for the agent. The position-to-direction map serves as the first stage of the transformation from desired location to motor command, and converts the allocentric position of the desired destination into an allocentric desired direction of movement by comparing the desired position with the current position of the animat. A desired direction ring compresses representation of the position-to-direction map into a one-dimensional allocentric desired direction vector. The direction-to-turn map then compares the allocentric desired direction with the allocentric current head direction and computes the egocentric turn force vector. To drive the vehicle, this signal can be interpreted as steering angle for the left or right turn.

[0108] *Spatial Planning*

[0109] SPARK' s spatial planning algorithm may be based on a neuronal parallel breadthfirst asymmetric bidirectional graph search where each location in the path planning system corresponds to a node in a graph. Edges in the graph may initially exist between neighboring locations, but as the agent explores the environment and encounters obstacles the corresponding edges may become impassable through obstacle learning and may be excluded from the breadth-first probing. This may result in a system that can adapt to found configurations of obstacles and find the substantially shortest path around these obstacles.

[0110] In general, the SPARK algorithm may model propagation of waves of activity from the goal location (reverse activity spread). In an unconstrained environment these wavefronts may form expanding concentric circles around the goal location. Wave propagation may not be allowed if an obstacle is encountered, so the wavefront deviates from the circle to flow around the obstacle as shown in Figure 7 by depicting obstacle map 700, wave propagation 702, and a combined view of both populations 704. As a result of all deviations, the first wavefront segments that reaches the current location of the agent comes from the direction that corresponds to the substantially shortest path to the goal. In response

NVIDIA Ex1002 Page 216 of 1081

to this, a new wavefront is initiated at the current location (forward activity spread). The collision of this front with the second wave coming from the goal may then happen at the location closest to the current location of the agent and in the direction of the shortest path towards the goal. This is called the next desired destination. Then the agent moves to this desired location and the process is repeated. A reset signal may be generated when the graph search finishes, which allows a new goal to potentially be selected as described above.

[0111] *Robotic implementation*

[0112] The above described navigation system may be implemented into a robotic system, autonomous agent, or other moving body. This section describes one exemplary implementation of the navigation system into an example robotic system.

[0113] The navigation system can be implemented into any microprocessor. In this exemplary implementation, the navigation system was implemented with Lego Mindstorms NXTs outfitted with 7 sensors and 4 motors. A Lego NXT is a microcontroller that can be interfaced with many different types of sensors and motors, all which require no soldering or significant setup time. The NXTs communicate with a low-power, low-cost single-board computer development platform (Pandaboard) which controls communications with a neural application client. The Pandaboard runs an asynchronous server that listens for commands from the neural application client while reading data from the NXTs and camera. When data is received from an NXT, it is repackaged and sent over UDP to the neural application client. The Pandaboard utilizes OpenCV to process image data from a USB camera and preprocesses/broadcasts it to neural application over TCP through an onboard router connected to the board through a Cat5e cable. The neural application client is capable of sending movement commands and can also schedule tasks on the NXTs.

[0114] Figure 8 shows a hardware diagram and the protocols used to connect the devices. In some implementations, a Lego Mindstorm Robot 800 may have left-front wheels 802, right-front wheels 804, back-left wheels 806, and back-right wheels 808. The robot may have sensors such as an RFID scanner 810, an EOPD sensor 812, an accelerometer 814, a front ultrasonic 816, a left ultrasonic 818, a right ultrasonic 820, and a gyroscope analog sensor 822. The sensors on the robot may be configured to communicate with a Pandaboard 824 configured with LCP over USB connectivity. The board may also support a webcam 826 and/or other camera and/or image-capturing device, as well as a communication device 828 configured to allow communications between the robot and another electronic device (e.g., a computer 830).

[0115] Figure 9 illustrates the main objects within the software of the system and their respective hardware hosts and communication streams. In some implementations, a PandaBoard 924 may be configured with a MasterCommandThread 900 which may send NXT Commands 902a and 902b to a Lego Mindstorm board configured with an NXTCommandReceiver 904a and 904b which may receive instructions and/or other data from the PandaBoard. Each robot may also be configured to support action loops 906a and 906b, which may include actions such as PackageSensors 912 (which in some implementations may be configured to interact with sensors 914 on the NXTController 922), and/or PackageDistanceSensors 908 (which may be configured to interact with DistanceSensors 910 on the NXTController). The PandaBoard may also be configured with an USBtoTCPCamera module 916 which may allow the PandaBoard to receive data from a webcam 918 and/or a similar image-capturing device and convert it into a stream over TCP protocol. The NXTController may also periodically send control commands 920 to the PandaBoard when necessary.

[0116] The robot was equipped with three ultrasonic sensors that detect objects in proximity. These readings are then used by the navigation system to form an obstacle map to avoid collisions. These sensors were placed on the right, left, and front of the robot. An accelerometer and gyroscope were used to detect the robots acceleration and angular momentum for use with the PInNAcLE subsystem. RFID tags and an RFID scanner emulated the more complex chemical scanners used in Mars Rovers. The RFID tags were placed around particular rocks, with each emitter tuned to a particular frequency. When a tag was detected, the RFID signal simulated a chemical detection of various aspects of rock geology (e.g., a particular chemical signature). The model reward system was then triggered and the rock was placed in the robot's internal map of the environment. These tags are also used by the robot to simulate recognition of the recharging base.

[0117] The robot experiments were conducted in a ~2x2 meters indoor simulation of the Martian landscape. This indoor facility provided a scaled-down, enclosed area with rocks and small terrain variation (the terrain is a tarp sheet with small elevation discontinuities) to test vision and navigation aspects of the navigation system.

[0118] *Model Performance*

[0119] The robot was tasked to explore the experimental environment; tracking its past and present position through the intelligent integration of self-velocity parameters given by inertial, motor, and a visual flow sense, learning the position of task-relevant objects such as

a home base, scientific targets, and obstacles; and managing competing drives for exploration and management of internal resources such as its battery level. The structure and physics of the virtual Mars environment were created and simulated with the JMonkey game engine, passed to the model using Virt-U package, and the neuromorphic algorithm controlling the virtual rover was simulated separately after having been written in the Cog Ex Machina simulator or JNDRA simulator (results not shown).

[0120] Figure 10 diagrams the specific phases of the main task. The virtual rover 1000 begins its exploration by leaving its base with full battery in order to begin exploring. (Figure 10, 1002). The rover explores unknown areas, maintaining its relative position and orientation with respect to the starting point and progressively building and updating its spatial representation of the environment mapping the obstacles 1004. This map is useful when the rover will, for instance, seek specific objects and locations in the environment or as a source for conversion into human readable format. After exploring for some time the rover's battery depletes 1006 to the point where it becomes motivated to return to the base 1008 where it began exploring by as direct of a route as possible (Figure 10, 1010). After recharging its battery the rover continues to explore from where it left off (Figure 10,1012). Figure 11 show the actual results in the virtual environment, and Figure 12 shows an incremental map view of the same trajectory (e.g., 1200 corresponds to path 1002, 1202 corresponds to path 1010, and 1204 corresponds to path 1012).

[0121] Figure 5 compares the compass heading of the physical rover through the simulation to heading determined by the integration of velocity information from one of several senses. Note that the final head direction estimate is determined by the integration of a combined estimate from all three senses as outlined above.

[0122] Autonomous exploration may include the management of multiple, overlapping, and oftentimes conflicting drives that keep the agent relatively safe and efficiently completing its task(s). The simulation requires the rover to balance its exploratory drive with a need to keep its battery charged; the balancing of goals by the navigation above. Figure 13 plots the rover's battery level over the course of the simulation 1302. The battery is empty at the beginning of the simulation 1304 in order to motivate the rover to learn the position of the charging base. Gating the learning of a reward by the desire for that reward may be a biologically realistic method of conserving memory and computational resources while retaining survival-relevant memories. When the battery is low enough, the rover revisits its base in order to recharge, as shown by the shaded areas 1306 and 1308 in Figure 13. The

NVIDIA Ex1002 Page 219 of 1081

rover takes longer to find the base on the second visit at minute twenty-three, both because the rover has traveled further from the base and because accumulated path integration error forces the rover to search in the immediate vicinity around its memorized position of the base. In both cases, however, the rover is able to find the charging base, recharge, and continue its exploration; the second decision to recharge is also shown in Figure 12 as the part of the trajectory stretching from about 0 m latitude to about 100 m latitude in the middle graph.

[0123] In addition to testing the performance of the navigation system in the virtual environment, the model was also demonstrated to operate on a physical robot. The construction of the robot and environment is detailed above. Figure 14 shows a short simulation of the robot moving in a preset circular arc. While traveling in this arc the robot moves over radio emitters that symbolize the charging base and a scientific target. In this simulation the navigation system's path integration (PInNAcLE) subsystem is driven exclusively by the motor system's preset velocity, sent as individual tread torque levels. The robotic platform is able to map out its course in the environment 1402, learn the positions of the same types of task-relevant objects as in the virtual environment 1404, and is able to recharge a simulated battery when moving over the charging base 1406.

[0124] The robot was also turned in place by setting the angular velocity to the sine of the simulation time scaled so that it reverses its direction of rotation around the vertical axis every twenty seconds. This preliminary test shows that the camera on the physical robot is able to accurately determine the robot's angular velocity at slow rotation speeds. In some implementations, increasing the camera's frame rate can increase at which speeds the navigation system can detect rotations. In some implementations, detection of rotation can be improved by smoothing the motion of the camera by adding a suspension system to the robot or by gating optic flow's influence on path integration by the other senses' estimates of rotation speed as disclosed above.

[0125] **OpenEye: An artificial visual system based on neural principles**

[0126] The OpenEye model is an active vision system that autonomously make decisions about where to look, generates saccades to target locations, and learn online view-invariant representations of salient objects. OpenEye models the parallel visual processing streams of mammalian visual cortex, and integrates in a single framework several capabilities that are typically modeled separately, such as the influence of bottom-up saliency and top-down spatial evidence on saccade decision-making, view-based object recognition, stable

NVIDIA Ex1002 Page 220 of 1081

incremental learning, and space-variant approaches to vision modeling In some implementations, the OpenEye and navigation system learns incrementally with a single presentation of the stimulus 1506, for example, it does not require a separate training and testing phase. OpenEye may also be able to operate with or without supervision.

[0127] In some implementations, OpenEye integrates bidirectional connections 1500 between dorsal (where) 1502 and ventral (what) visual processing streams 1504 (e.g., see Figure 15). OpenEye's where pathway may use simulated eye movements to navigate through a map of candidate areas established by bottom-up surface properties and top-down spatial evidence in order to explore distinct views of each object. By using a log-polar representation of the retinal visual field, OpenEye may sample the image "view" generated by each saccade in a space-variant manner such that an object in the center of the system's visual field (the fovea) is represented in high resolution while objects in the periphery are represented with nonlinearly decreasing resolution. This may save a processing power, as, for example, an image field of view can be processed in an order of magnitude smaller neural population using the log-polar compression.

[0128] Once the where pathway has chosen a point of foveation, it forms a surface-fitting attentional shroud around the object in order to isolate it from other surrounding objects. This is implemented using feedback principle outlined above. The attentional shroud may provide OpenEye with the capability to discover whether sequential views belong to the same or to different objects. The what pathway may be modeled as a series of hierarchical classifiers that cluster objects into view, object, and/or name categories. After the what pathway has identified an object, the what-to-where feedback may use learned prototypical object-feature maps to produce "smart saccades" in order to focus on spatial locations that yield greatest disambiguation of competing object identity.

[0129] The Martian Virtual Environment used to train OpenEye was created using the jMonkey 3 graphical engine and interfaced with the neural model using Virt_U package. OpenEye was trained to recognize two types of Martian rocks (type 1 1600 and type 2 1602) in Figure 16 with a supervising name signal for 250 cycles. The scene is then switched to a similar scenario, but this time, there was no supervision, and there was a third, novel type of rock that the system has not been trained on. The accuracy of classification in this testing scene is close to 100%. At first, the system is not able to recognize the novel rocks, and there is no name associated with the newly created object. This triggers a novelty signal in OpenEye, which can be returned to the user as an alert, or be utilized to capture snapshots of

NVIDIA Ex1002 Page 221 of 1081

the novel rock for closer examination. This capability allows the system to generate alerts and capture views 1604 not only of known science targets, but also novel targets 1606 of potential interest to the scientist. Eventually, once OpenEye has learned several views of these novel objects, it is able to classify them properly upon subsequent foveation, and the rock becomes part of the "object dictionary" of OpenEye, where a user can choose to attach a verbal label to the rock.

[0130] Virt-U: Apparatus and methods to connect off-the-shelf 3D engines or physical robots with neural simulation software packages.

[0131] The system disclosed herein includes a virtual universe (Virt-U) software component 1700 designed to simplify prototyping of neural models of robotic intelligence and test them in a virtual environment before implementing on physical robotic platform. With Virt-U a user can connect an existing neural modeling software package with another existing 3D simulation engine, for example a game engine.

[0132] In this scheme virtual environments 1702 supply rich sensory and proprioceptive signals attached to an agent, while the neural model 1704 (also referred to as the brain) receives and processes sensory data to produce motor control commands required to elicit the agent's corresponding behavior (e.g., see Figure 17).

[0133] The example system binds to a neural engine of choice that may allow for neural models to be executed in a substantially parallelized, distributed computation environment possibly on one or more interconnected computer systems with GPUs. In two example instantiations we connected Virt_U to Cog Ex Machina software from Hewlett Packard, and to INDRA engine developed by Neurala Inc. Virt-U then delivers sensory data to the neural model and behavioral commands (translated to physics and graphics) to the 3D engine. The same API that Virt-U exposes to neural engine for connectivity with the virtual environment is used to connect neural engine with the robotic platform.

[0134] *Virt-U Architectural Principles*

[0135] Virt-U architectural principles abstract the interface that connects a virtual world or robotic platform with its neural modeling environment. In order to segregate dependencies of the neural modeling environment from those of the virtual environment engine the Virt-U architecture includes of two major layers: the VE layer and the proxy layer. The VE layer abstracts the virtual environment engine and ads functionality that for example allows extracting sensory information that is not usually available from the game engine. The proxy layer delivers this sensory information to the neural model and extracts neural information

NVIDIA Ex1002 Page 222 of 1081

from the neural model in order to perform the behavior. The Virt-U can have two modes, particularly: 1. Simulation - when the behavior of the animat is controlled by the brain, and 2. No-brainer - no brain is attached, used for testing virtual environments.

[0136] Virt-U considers the world to be populated by virtual objects. These virtual objects can be animated (brained) and non-animated (brainless). Brained objects, called animats, can be controlled by neural model. An animat's brains receives sensory (e.g., visual) and/or proprioceptive signals and can then exhibit a behavior that follows the laws of physics to a predefined level. Animats can be either virtual animals (e.g., a rat) or vehicles (e.g., a rover), depending on how their motor functions are controlled, but a brain in both cases controls behavior. Brainless objects, called items, are entirely controlled by the virtual environment and obey its physics. Items include rocks, buildings, trees, etc. and can be considered a source for sensory information including rewards sensed by animats.

[0137] Virt-U considers an animat to be a collection of sensory organs and animat controls. Sensory organs may in turn include sub-sensory organs, which ultimately can be connected with virtual sensors dedicated to collecting sensory information. For example, an animat may have a sensory organ called "eyes" that comprises individual eyes, where each eye contains facets connected with virtual cameras. All sensory organs are responsible for maintaining and updating sensory information for the sub-organs they contain thus allowing for recursive sensory data update to be performed on an animat. Animat controls constitute outgoing interfaces that may be accessed by the brain to perform the predetermined behavior. Sensory organs and animat controls expose incoming and outgoing interfaces to be accessed from the Virt-U proxy layer in order to supply and deliver sensory and motor control information.

[0138] On the neural modeling side (brain side), proxy organ controllers perform access to the VE. These controller can retrieve and supply neural data by accessing a specific organ controllers' incoming and outgoing interfaces. Proxy controllers can abstract internal knowledge of sensory organs from the neural model by acting as hardware drivers within the neural modeling environment. In simulation mode, Virt-U can create the virtual world based on an XML description. This description is sufficient to define the world, the animats with all their sensory organs and controls, and all virtual objects with their associated rewards. This XML-based virtual world description ensures Virt-U portability between various neural modeling environments.

[0139] *Virt-U VE Layer for JMonkey Game Engine*

NVIDIA Ex1002 Page 223 of 1081

[0140] An example implementation of the Virt-U VE layer presented here complements the open source JMonkey game engine.

[0141] *Virt-U Application and Library*

[0142] The Virt-U library platform may be a general-purpose, Java-based game development environment that extends 3D graphics game development capabilities to the Java programming community. The Virt-U may include a system that maintains a variety of assets, for example, models, textures, sounds, whole scenes, rendering of 3D graphics, and the ability to simulate physical interactions.

[0143] The Virt-U library layer may isolate the development of a brain-controlled animat from the specifics and complexity related to 3D game programming. On the user level, a user may create and run a virtual world populated by an animat by writing Java code, instantiating the classes representing the world and the preexisting animat, and adding external cameras to the world as necessary. In some implementations, once the classes are instantiated, the VE is run by the JMonkey game engine and can be controlled via the JMonkey GUI.

[0144] In some implementations, Virt-U application is a platform application that builds custom virtual worlds at runtime, which extends the DebugWorld class and implements the WorldBuilder interface comprising a single construct method. Virt-U application is embodied in DebugApplication - an executable intended for testing custom virtual environments in the no-brainer mode and generation of world XML-based description to be used in simulation mode.

[0145] As a command line argument, Virt-U application might take a name of a custom class that extends DebugWorld. An example of such a class is presented below.

[0146] Once the construct method yields, the debug application runs under control of the physics engine and can be interacted with via JMonkey GUI, which interprets keyboard and mouse actions as animat movement commands. The application also extends the Virt-U command set that allows switching between external cameras views and first-person views from the perspective of different animats. In some implementations, the Virt-U Builder can have one or more user customizable features, which can include:

[0147] a custom set of VE assets that determine how each individual object looks to the observer and allows for positioning of these objects in the scene;

[0148] different pre-created animats placed at specific locations in the VE scene; and

[0149] placement of virtual cameras characterized by a custom set of parameters to achieve a third-person view of certain portions of the scene.

NVIDIA Ex1002 Page 224 of 1081

[0150] *Running Virt-U Application in the No-brainer Mode*

[0151] In some implementations, the physics engine runs VE applications on its internal clock that synchronizes physical interactions with 3D graphic rendering. Users can control execution of the Virt-U application in no-brainer mode via the GUI described in Table 1.1 below. In other implementations Virt-U library exposes all the necessary API to an external application and provides it with all the necessary functionality to establish a similar GUI bypassing Virt-U application.

[0152] *Virt-U Implementation*

[0153] Implementation of Virt-U may comprise Java classes (e.g., see Figure 18) and interfaces that are extended or implemented to abstract the functionality to supply sensory information to the neural model as well as perform a behavior based on exposed motor control interfaces. The following sections discuss example Virt-U classes.

[0154] *VirtUApplication Class*

[0155] The VirtUApplication class 1800 is a focal point of communication between the game engine and Virt-U. The class implements JMonkey's SystemListener interface and thus provides an entry point for handling GUI events, rendering graphics, updating sensory information, and/or exercising controlled behavior. The VirtUApplication class initializes components of the JMonkey game engine, creates a rendering context, and instantiates the world. When JMonkey Engine calls the update method of the SystemListener interface in the VirtUApplication class, the latter may call the update method on the World class. In turn, the World class may call the update method on all objects populating the world.

[0156] VirtUApplication is extended by the DebugApplication class that is part of Virt-U application and runs Virt-U in no-brainer mode. In simulation mode, the VirtUAplication class may be extended by different application classes that expose the engine in different modes: synchronized to the neural simulation clock or running asynchronously. On the neural simulation proxy level these classes can be further extended or wrapped depending on the neural engine implementation. In one example of the proxy layer the CogVirtUApplication provides an entry point to control execution of Virt-U from Cog Ex Machina package written in Scala. In another implementation VirtUAplication is wrapped in Java Native Interface object to provide an entry point to the engine from INDRA package written in C++. Note that Virt-UApplication is a singleton class and, in described implementations, only one instance of VirtUApplication can exist.

[0157] *World Class*

NVIDIA Ex1002 Page 225 of 1081

[0158] The World class 1802 serves as the main container for landscape, animats, items, and external cameras. The world appearance is determined by the scene model specified as an argument to the addScene method. The scene is encoded by the model's .j3o file that can be generated by Virt-U application or outside of Virt-U using third party software. In some implementations, by default, Virt-U adds an external chase camera positioned at a predetermined distance close to the animat. World is a singleton class. Animats may be added by the addAnimat method that takes a reference to the constructed animat and coordinates specification of an animat's initial location. External objects may be added to the world by calling addObject with an instance of the item class passed as an argument. External cameras may be added to the world instance by calling addCamera with arguments specifying location, orientation, and frustum of the camera.

[0159] The World class instance may be responsible for updating the state of animated and non-animated objects it contains. This may be performed by calling the update method on these objects. Similarly the update method is called on each non-animated objects. The World singleton object also calls the reset method for animated and no-animated objects it contains in a manner similar to that of update.

[0160] World class itself exposes the update and reset method called internally by the VirtUApplication object which exhibits SystemListener interface and thus is called by the engine on each step to update world's state and the state of its components.

[0161] *Animat Class*

[0162] The Animat class 1804 can serve as the base class for implementation of animats with a desired set of properties. Animat implementation details are described later in this report. In some implementations, another function of the Animat class is to serve as a container for sensory organs and animat control. The Animat class implements the update method from which it calls update for each sensory organ that the animat contains. Sensory organs may be added to the animat during its construction or initialization by calling an addController method that takes OrganController 1806 reference as an argument.

[0163] To control animat behavior, a control object may be created. The control object is animat-specific and depends on the type of outgoing interfaces exposed by the ultimate animat class. For example, walking animats such as a virtual animal expose the walking controls, allowing the caller to specify the direction and speed of walking or running. Vehicle animats expose vehicle controls, allowing the caller to specify acceleration and steering. Control objects are added via an addControl method typically called during animat

NVIDIA Ex1002 Page 226 of 1081

initialization or construction, passing a control object reference as an argument. All elements of an animat's behavioral interface may be accessed via its control object.

[0164] In some implementations, another function of the Animat class is to generate and hold a collision shape. The collision shape may be generated in the shape of a capsule surrounding the animat (depending on its dimensions) or as a more tightly fit shape following the animat shape. The collision shape serves as a proximity sensor as well as a chase camera boundary.

[0165] *Vehicle Classes*

[0166] The Vehicle class is derived from the animat class to encapsulate functionality related to vehicle control and vehicle composition. The specific vehicle class may be derived from the Vehicle class. The specific vehicle class may determine the appearance and navigational properties of the vehicle, sets up relevant sensory organs and motor controls, and determines its sensitivity to reward values as well as its collision properties. The Vehicle class may receive and interpret a user's input for control of the animat in no-brainer mode. The user's input may allow the user to accelerate/decelerate and steer the vehicle.

[0167] *Other Animat Classes*

[0168] The Virt-U may also include the CharacterAnimat and specific character classes that implement a walking character whose motion is based entirely on kinematics. A walking character may be controlled by standard GUI commands that allow the user manipulate the direction of motion (forward/backward, left/right turns, jumping).

[0169] *OrganController Class*

[0170] OrganController may serve as the base class for all sensory organs. It may define the update and reset methods 1812 that may be implemented by the derived classes. In some implementations, the purpose of classes that extend OrganController is either to supply sensory information specific to an animat's senses 1808a, 1810a or to contain other sensory organs 1808b, 1810b.

[0171] *OculoMotor Class*

[0172] This class handles visual inputs from single- or multi-camera sources as well as rotation of the cameras. Multi-camera sources are used here to simulate, with minimal distortion, large (>75 degrees) fields of view (FoV) that are attainable with certain real world cameras. In some embodiments, the multi-camera source works by combining inputs from several narrow FoV cameras (or facets) and deliver the result of this operation as a single

NVIDIA Ex1002 Page 227 of 1081

input data stream. The OculoMotor class acts as a container for Eye class objects, which in turn contain camera sensor facets.

[0173] *Vestibular Class*

[0174] This class may embody vestibular inputs that record self-motion of the agent. The vestibular sensors may provide the model with linear and angular velocities or accelerations of the agent as would be done if the respective accelerations were measured by accelerometers and gyroscopes of the real robotic device. Realistic noise models can be added to further improve the quality of simulated by Vestibular class real IMUs.

[0175] *Rewards and Collisions*

[0176] In some implementations, rewards and collisions may be accessed through their respective controllers to obtain reward and collision or touch sense sensory data. The update of touch and reward information may happen due to implementation of the CollisionListener interface by their respective animat classes. Collision may be detected via a collision body: an invisible and weightless shape surrounding the animat that serves as a proximity sensor. Collision or detection of an external object by proximity sensor can result in obtaining a reward value if a reward is attached to an object that the animate collides with or senses via proximity sensor. To sense a reward, the animat should have a corresponding reward sense. Sensing the reward information allows reinforcement learning to be implemented as part of the neural model controlling animat behavior. Lastly, a stationary object that extends the Item class can be associated with different rewards types holding a different reward value.

[0177] *TouchSense Class*

[0178] TouchSense abstracts touch/proximity sensors that provide a vector in polar coordinates centered on the agent's center of mass and pointing to the locations of collision (for touch sensor) or close-to-collision (proximity sensor). In some implementations, a difference between touch and proximity is the fitness of the collision surface for the agent. For example, for a touch sensor the collision surface substantially conforms to the shape of the body, whereas a shape surrounding the body with a radius corresponding to the sensor range is created for proximity sensors. In the touch sensor the length of the vector can relate to the strength of an impact, and in the proximity sensor the vector length can relate to the distance between the agent and the edge of an obstacle. In some implementations, multiple collisions can be detected and passed to the model simultaneously.

[0179] *RewardSense Class*

[0180] RewardSense abstracts an arbitrary type of reward in order to model positive or negative reinforcement. In the context of Mars exploration, RewardSense sensors can simulate results coming from chemical analysis of samples collected by the Rover. Any object in the VE can be assigned a reward type and value by the VE designer, and if the animat comes in contact with such an object during the simulation it will receive information about the associated reward type and its value.

[0181] *Other Sensory Organ Classes*

[0182] In some implementations, the Virt-U system can include additional sensors for testing and verification. These included a GPS-like sensor that provides the true coordinates of an agent at any point in time. Such a sensor could be used to verify and tune precision of the internal path integration system.

[0183] *AnimatVehicle Class*

[0184] The AnimatVehicle class provides entry points to control a vehicle that can use the JBullet ray-casting mechanism to mimic suspension of a physical vehicle. This class extends the JMonkey VehicleControl class and implements the CollisionListener interface. The latter constitutes a callback from the JMonkeyEngine that is called to handle collisions. The VehicleAnimat class can act asynchronously compared to the JMonkey game engine clock via the GUI interface, but actions can occur on game engine clock ticks. The update method implemented by the AnimatVehicle Class can execute queued control commands to the vehicle.

[0185] The collision method called back by JMonkeyEngine to handle collision events may allow the AnimatVehicle class to populate the touch or proximity sensor map. The touch map can be retrieved upon request from the TouchSense organ, but actual touch map generation can occur within the AnimatVehicle class. The reward may be updated if a collision body other than the animat has a reward type for which the animat subscribes by having an attached reward sense. Reward information (different from collision information) related to the nature, or quality, of the reward can also be filtered for environment.

[0186] *AnimatCharacter Class*

[0187] The AnimatCharacter class may provide entry points to control a walking character. The class extends JMonkey's PhysicalCollisionObject class that ultimately implements the CollisionListener interface. Actual movement control and collision handling may be implemented in the aggregated AnimatCharacterControlier object which implements the ActionListener interface. AnimatCharacterController controls movement based on linear

NVIDIA Ex1002 Page 229 of 1081

and angular velocities, and supports movement along an animat's own forward/backward and left/right axes as well as world up/down axis plus rotations of heading and elevation. Collisions may make the animat slide along the colliding surface by changing heading and/or elevation. AnimatCharacterController may also use the ghost object and convex sweep test to search for upcoming collisions. This may be combined with discrete collision detection to recover from collisions with objects. Lastly, AnimatCharacterController handles collision and reward information similarly to the manner described above for AnimatVehicle objects.

[0188] *Externally driven rendering context*

[0189] The physical clock that exists in JMonkey engine operates in units of time that corresponds to the physical world where objects have simulated masses, accelerations, torques, forces, and other physical properties in corresponding units. The graphical clock in the engine is synchronized to the physical clock, yet both clocks are designed to run asynchronously based on user input. In the case when neural simulation speed differs significantly between neural simulator and robotic embodiment this may not be appropriate, and the Virt-U system may synchronize the simulation clock with these two engine clocks so that inputs and outputs between the brain model and the animat's body operate on the same clock simulating the real time interactions. To achieve this synchronization, a custom rendering context was created that is driven by the simulation clock. On every tick of the simulation clock the world advances for a time interval that is equal to the length of the model processing cycle.

[0190] *Virt-U Proxy Layers for Cog Ex Machina and INDRA neural simulators.*

[0191] Virt-U proxy layer may expose Virt-U library interface to neural simulation software. Depending on the programming language used in the neural simulator, Virt-U proxy layer may be a collection of Java and/or Scala classes (example implementation for Cog Ex Machina) or a collection of C++ classes with internal Java Native Interface wrappers around Virt-U library calls (example implementation for C++ based INDRA neural simulator).

[0192] In some embodiments, the Virt-U proxy layer for Cog Ex Machina (e.g., see Figure 19) is designed for dynamic update and extraction of designated sensory (1904 to 1900, to 1906 to 1902) and motor control (1902 to 1910 to 1904) dynamic fields in the neural model (1908). Cog Ex Machina adopts a synchronous and entirely deterministic model of computation that may define the computational process as a sequence of steps. Updates occur

NVIDIA Ex1002 Page 230 of 1081

substantially simultaneously after execution of each step - this includes sensory information injection and motor control information extraction.

[0193] INDRA based proxy layer adopts an asynchronous model between virtual environment and neural model. This allows a user to test the performance of complicated neural models in real time and simplifies their transition to robotic embodiments. Virt-U proxy layer for INDRA may add time-averaging to the sensory readings for the cases when virtual environment queries sensors more often than the neural model can process the information.

[0194] *Manipulating Virt-U from Neural Application*

[0195] The neural application initializes a VE or World 1914 using a path to the XML description 1912 of the virtual world. This may create neural populations 1908 before the virtual world is instantiated and its sensory organs become accessible like in Cog Ex Machina proxy. This may also follow the opposite order and instantiate virtual world 1904 and sensory/motor buffers 1906 before any neural populations are created as in iNDRA proxy. In both cases the subsequent construction within the proxy instantiates the missing parts and links them together. In some cases this linkage can be established by providing pointers to the neural data structures to the respective sensory/motor objects of the proxy 1900. In other cases it can be established by providing callbacks 1906 that execute sensory motor operations on the neural populations.

[0196] The mechanism by which the outside world connects to the neural model is called injection/extraction. Using injection, data can be placed into a neural population; extraction may allow data to be pulled from the neural population to perform a desired behavior.

[0197] *Proxy Controllers*

[0198] The proxy controllers 1900 serve as either input or output controllers between Virt-U proxy layer and Virt-U library. Actual data exchange occurs in the update method. Input controllers may acquire data from Virt-U sensory organs and inject them into virtual sensors. Output controllers may acquire data from virtual actuators and use animat control interfaces to perform the animat's desired behavior.

[0199] DebugWorld example class

```
public class TestMarsWorld extends DebugWorld implements WorldBuilder
{
 //...
  super();
```

```
  app.setDebugCollisionShapes(true);
}
@Override
public void construct()
{
World.instance.addScene("Scenes/Mars.j3o");
        Rover animat = new Rover("Rover", "Models/RoverParts/RoverBody.j3o");
        World.instance.addAnimat(animat, new Vector3f(0.0f, 93f, 0f), 2.0f*((float)
Math.PI)/3.0f, 0.0f);
        // add cameras
}
```

[0200] **GUI commands for the Virt-U Builder application**

User Action	System response
Key H	For animals: Set constant speed to move the active animat to the left. Speed remains constant until a conflicting command is issued. For vehicles: Apply steering to the left. Steering is only applied while the key is pressed.
Key K	For animals: Set constant speed to move the active animat to the right. Speed remains constant until a conflicting command is issued. For vehicles: Apply steering to the right. Steering is only applied while the key is pressed.
Key U	For animals: Set constant speed to move the active animat forward. Speed remains constant until a conflicting command is issued. For vehicles: Apply forward acceleration. Releasing the key leaves the speed constant (pending terrain slopes and obstacles).
Key J	For animals: Set constant speed to move the active animat backward. Speed remains constant until a conflicting command is issued. For vehicles: Apply brakes. Releasing the key releases the brake.
Key G	For animals: Set constant speed to move the active animat upwards. Speed remains constant until a conflicting command is issued. For vehicles: No effect.
Key B	For animals: Set constant speed to move the active animat downwards. Speed remains constant until a conflicting command is issued. For vehicles: No effect.
Spacebar Key	For animals: Set all speeds to zero. For vehicles: Jump in the air.
Key comma (,)	For animals: Set constant angular velocity of the active animat rotating it to the left. Velocity remains constant until a conflicting command is issued. For vehicle: No effect.
Key period (.)	For animals: Set constant angular velocity of the active animat rotating it to the left. Velocity remains constant until a conflicting command is issued. For vehicle: No effect.
Key F	For animals: Set constant angular velocity of the active animat rotating its front upward. Velocity remains constant until a conflicting command is issued. For vehicle: No effect.

User Action	System response
Key V	For animals: Set constant angular velocity of the active animat rotating its front downward. Velocity remains constant until a conflicting command is issued. For vehicle: no effect.
Key E	Switch to first person view from the eye of the active animat.
Key [	Each press cycles through available eyes of the active animat, making the next eye the source of the first person view.
Keys 0-9	Switch the active external camera-to-camera. Camera 0 is a follow camera that always has the first animat in focus. Cameras 1-9 are optional cameras that are fixed in space. If the world has less than 9 fixed cameras the extra keys have no effect.
Mouse move	No effect if the camera 0 or first person view is active. Rotate the active camera if camera 1-9 is active.
Mouse button press + mouse move	If camera 0 is active, rotate the camera view.
Mouse wheel	Zoom-in/zoom-out with the camera.
Key C	Outputs the current position and direction of the active camera. This is useful for precise camera positioning.
Keys W,S,A,D	Move cameras 1-9 forward, backward, left, and right, respectively.
Keys Q,Z	Move cameras 1-9 upward and downward, respectively
Keys F1-F10	Set the active animat to animat 1-10. If there are fewer than 10 animats the extra keys have no effect.
Key TAB	Saves the current world configuration to an XML file.
Key P	Pauses and unpauses the rendering.
Escape Key	Exits the application.
Key M	Outputs memory usage of the application to the standard output.

Table I.1. GUI commands for the Virt-U Builder application

[0201] **List of Virt-U sensors-actuators**

- VirtualSensor
- VirtualVectorSensor
- VirtualColorSensor
- VirtualRetina
- VirtualColorRetina
- VirtualScalarFieldActuator
- Virtual0DSensor
- Virtual0DScalarFieldActuator

[0202] **Proxy controllers implemented to support simulation during Phase I**

- **StaticCameraIC** – injects color images from static cameras

NVIDIA Ex1002 Page 233 of 1081

- **EyesIC** – injects color input images from virtual vision sensory organs;
- **VestibularIC** – injects vestibular data in three 0D scalar fields;
- **RewardsIC** – injects reward values into an array of 0D sensors;
- **TouchIC** – injects touch information into a 2D scalar field representing a touch map in polar coordinates;
- **VehicleLinearSpeedIC** –injects vehicle linear velocity into a 0D scalar field;
- **VehicleAngularVelocityIC** – injects vehicle angular velocity into a 0D scalar field;
- **GlobalPositionIC** – injects GPS data into a 0D vector field;
- **EyesOC**–extracts eye movement information from the actuator field; and
- **DualMotorOC** – enforces vehicle angular and linear velocity based on values extracted from two 0D dynamic scalar fields.

[0203] **Example of Cog Virt-U application (animat wandering a virtual world)**

```
object Wandering extends CogBrowserApp("Wandering around", SimConfig.rate)
{
  val engine = new CogVirtUEngine("assets/Worlds/TestWorld", setupControllers)

  //The rest of the object is your network

  //create sensor and actuator fields
  val otolithX = new Virtual0DSensor("otolith X")
  val otolithZ = new Virtual0DSensor("otolith Z")
  val sscanal = new Virtual0DSensor("sscanal")
  val gps = new VirtualVectorSensor(3,"GPS")
  val somatoSensorSize = 100
  val touch = new VirtualSensor(somatoSensorSize,  somatoSensorSize,"touch")
  val rewards = new Virtual0DSensor("comfort")
  val staticCameras = new StaticCameraFields
  val lv = new PresetLVField("LinearVelocity", 0.4f)
  val av = new PresetAVField("AngularVelocity", 30)
  val rmmcf = new RandomMicroMacroSaccadesField("EyeRotation", .01f*SimConfig.rate,
.01f*SimConfig.rate, 30)
  val eye = new FovealField("Rat","Right Eye")

  //brain model ……

  //callback to initialize sensor/actuator controllers
  def setupControllers {
    val world = World.instance//engine.getWorld
    engine.registerController(new StaticCameraIC( world.cameras(),staticCameras))
    val animat = world.animats().get(0)
    engine.registerController(new EyesIC(animat,eye))
    engine.registerController(new VestibularIC(animat,otolithX,otolithZ,sscanal))
    engine.registerController(new GlobalPositionIC(animat,gps))
    engine.registerController(new RewardsIC(animat,rewards))
    engine.registerController(new TouchIC(animat,touch))
    engine.registerController(new EyesOC(animat,rmmcf.saccadeField))
```

```
    engine.registerController(new
DualMotorOC(animat.asInstanceOf[CharacterAnimat],av.avField,lv.lvField, false))
  }
}
```

[0204] **Conclusion**

[0205] While various inventive embodiments have been described and illustrated herein, those of ordinary skill in the art will readily envision a variety of other means and/or structures for performing the function and/or obtaining the results and/or one or more of the advantages described herein, and each of such variations and/or modifications is deemed to be within the scope of the inventive embodiments described herein. More generally, those skilled in the art will readily appreciate that all parameters, dimensions, materials, and configurations described herein are meant to be exemplary and that the actual parameters, dimensions, materials, and/or configurations will depend upon the specific application or applications for which the inventive teachings is/are used. Those skilled in the art will recognize, or be able to ascertain using no more than routine experimentation, many equivalents to the specific inventive embodiments described herein. It is, therefore, to be understood that the foregoing embodiments are presented by way of example only and that, within the scope of the appended claims and equivalents thereto, inventive embodiments may be practiced otherwise than as specifically described and claimed. Inventive embodiments of the present disclosure are directed to each individual feature, system, article, material, kit, and/or method described herein. In addition, any combination of two or more such features, systems, articles, materials, kits, and/or methods, if such features, systems, articles, materials, kits, and/or methods are not mutually inconsistent, is included within the inventive scope of the present disclosure.

[0206] The above-described embodiments can be implemented in any of numerous ways. For example, embodiments of designing and making the coupling structures and diffractive optical elements disclosed herein may be implemented using hardware, software or a combination thereof. When implemented in software, the software code can be executed on any suitable processor or collection of processors, whether provided in a single computer or distributed among multiple computers.

[0207] Further, it should be appreciated that a computer may be embodied in any of a number of forms, such as a rack-mounted computer, a desktop computer, a laptop computer, or a tablet computer. Additionally, a computer may be embedded in a device not generally

NVIDIA Ex1002 Page 235 of 1081

regarded as a computer but with suitable processing capabilities, including a Personal Digital Assistant (PDA), a smart phone or any other suitable portable or fixed electronic device.

[0208] Also, a computer may have one or more input and output devices. These devices can be used, among other things, to present a user interface. Examples of output devices that can be used to provide a user interface include printers or display screens for visual presentation of output and speakers or other sound generating devices for audible presentation of output. Examples of input devices that can be used for a user interface include keyboards, and pointing devices, such as mice, touch pads, and digitizing tablets. As another example, a computer may receive input information through speech recognition or in other audible format.

[0209] Such computers may be interconnected by one or more networks in any suitable form, including a local area network or a wide area network, such as an enterprise network, and intelligent network (IN) or the Internet. Such networks may be based on any suitable technology and may operate according to any suitable protocol and may include wireless networks, wired networks or fiber optic networks.

[0210] The various methods or processes (e.g., of designing and making the coupling structures and diffractive optical elements disclosed above) outlined herein may be coded as software that is executable on one or more processors that employ any one of a variety of operating systems or platforms. Additionally, such software may be written using any of a number of suitable programming languages and/or programming or scripting tools, and also may be compiled as executable machine language code or intermediate code that is executed on a framework or virtual machine.

[0211] In this respect, various inventive concepts may be embodied as a computer readable storage medium (or multiple computer readable storage media) (e.g., a computer memory, one or more floppy discs, compact discs, optical discs, magnetic tapes, flash memories, circuit configurations in Field Programmable Gate Arrays or other semiconductor devices, or other non-transitory medium or tangible computer storage medium) encoded with one or more programs that, when executed on one or more computers or other processors, perform methods that implement the various embodiments of the invention discussed above. The computer readable medium or media can be transportable, such that the program or programs stored thereon can be loaded onto one or more different computers or other processors to implement various aspects of the present invention as discussed above.

[0212] The terms "program" or "software" are used herein in a generic sense to refer to any type of computer code or set of computer-executable instructions that can be employed to program a computer or other processor to implement various aspects of embodiments as discussed above. Additionally, it should be appreciated that according to one aspect, one or more computer programs that when executed perform methods of the present invention need not reside on a single computer or processor, but may be distributed in a modular fashion amongst a number of different computers or processors to implement various aspects of the present invention.

[0213] Computer-executable instructions may be in many forms, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically the functionality of the program modules may be combined or distributed as desired in various embodiments.

[0214] Also, data structures may be stored in computer-readable media in any suitable form. For simplicity of illustration, data structures may be shown to have fields that are related through location in the data structure. Such relationships may likewise be achieved by assigning storage for the fields with locations in a computer-readable medium that convey relationship between the fields. However, any suitable mechanism may be used to establish a relationship between information in fields of a data structure, including through the use of pointers, tags or other mechanisms that establish relationship between data elements.

[0215] Also, various inventive concepts may be embodied as one or more methods, of which an example has been provided. The acts performed as part of the method may be ordered in any suitable way. Accordingly, embodiments may be constructed in which acts are performed in an order different than illustrated, which may include performing some acts simultaneously, even though shown as sequential acts in illustrative embodiments.

[0216] All definitions, as defined and used herein, should be understood to control over dictionary definitions, definitions in documents incorporated by reference, and/or ordinary meanings of the defined terms.

[0217] The indefinite articles "a" and "an," as used herein in the specification and in the claims, unless clearly indicated to the contrary, should be understood to mean "at least one."

[0218] The phrase "and/or," as used herein in the specification and in the claims, should be understood to mean "either or both" of the elements so conjoined, i.e., elements that are

conjunctively present in some cases and disjunctively present in other cases. Multiple elements listed with "and/or" should be construed in the same fashion, i.e., "one or more" of the elements so conjoined. Other elements may optionally be present other than the elements specifically identified by the "and/or" clause, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, a reference to "A and/or B", when used in conjunction with open-ended language such as "comprising" can refer, in one embodiment, to A only (optionally including elements other than B); in another embodiment, to B only (optionally including elements other than A); in yet another embodiment, to both A and B (optionally including other elements); etc.

[0219] As used herein in the specification and in the claims, "or" should be understood to have the same meaning as "and/or" as defined above. For example, when separating items in a list, "or" or "and/or" shall be interpreted as being inclusive, i.e., the inclusion of at least one, but also including more than one, of a number or list of elements, and, optionally, additional unlisted items. Only terms clearly indicated to the contrary, such as "only one of" or "exactly one of," or, when used in the claims, "consisting of," will refer to the inclusion of exactly one element of a number or list of elements. In general, the term "or" as used herein shall only be interpreted as indicating exclusive alternatives (i.e. "one or the other but not both") when preceded by terms of exclusivity, such as "either," "one of," "only one of," or "exactly one of." "Consisting essentially of," when used in the claims, shall have its ordinary meaning as used in the field of patent law.

[0220] As used herein in the specification and in the claims, the phrase "at least one," in reference to a list of one or more elements, should be understood to mean at least one element selected from any one or more of the elements in the list of elements, but not necessarily including at least one of each and every element specifically listed within the list of elements and not excluding any combinations of elements in the list of elements. This definition also allows that elements may optionally be present other than the elements specifically identified within the list of elements to which the phrase "at least one" refers, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, "at least one of A and B" (or, equivalently, "at least one of A or B," or, equivalently "at least one of A and/or B") can refer, in one embodiment, to at least one, optionally including more than one, A, with no B present (and optionally including elements other than B); in another embodiment, to at least one, optionally including more than one, B, with no A present (and

optionally including elements other than A); in yet another embodiment, to at least one, optionally including more than one, A, and at least one, optionally including more than one, B (and optionally including other elements); etc.

[0221] In the claims, as well as in the specification above, all transitional phrases such as "comprising," "including," "carrying," "having," "containing," "involving," "holding," "composed of," and the like are to be understood to be open-ended, i.e., to mean including but not limited to. Only the transitional phrases "consisting of" and "consisting essentially of" shall be closed or semi-closed transitional phrases, respectively, as set forth in the United States Patent Office Manual of Patent Examining Procedures, Section 2111.03.

CLAIMS

1. A method for aggregating data representing a moving body with respect to an environment, the method comprising:

(A) receiving, by a processor, a plurality of sets of data readings from at least one sensor, the plurality of sets of data readings representing independent measurements by the at least one sensor of the moving body with respect to the environment;

(B) mapping the plurality of sets of data readings received in (A) to respective first sets of data cells, each first set of data cells forming a corresponding first neural representation of the moving body with respect to the environment;

(C) applying respective weights to values in the respective first sets of data cells mapped in (B), the respective weights representing respective precisions of the independent measurements represented by the corresponding sets of data readings;

(D) combining the values in the first sets of data cells weighted in (C) to form a plurality of aggregated data readings; and

(E) mapping the plurality of aggregated data readings formed in (D) to a second set of data cells stored in the memory, the second set of data cells forming a second neural representation of the moving body with respect to the environment that is more precise than the first neural representations of the moving body with respect to the environment.

2. The method of claim 1, wherein (A) comprises receiving data readings representative of an orientation, motion, and/or position of the moving body with respect to the environment.

3. The method of claim 1, wherein (A) comprises processing data from a single sensor to form the first plurality of data readings and the second plurality of data readings by extracting different modalities of data reading from the data from the single sensor.

4. The method of claim 1, wherein (A) comprises receiving the first plurality of data readings from a first sensor and the second plurality of data readings from a second sensor.

5. The method of claim 1, wherein (A) comprises receiving the first plurality of data readings from at least one of a motor, a motor controller, an accelerometer, a gyroscope, an inertial measurement unit, an imaging sensor, a proximity sensor, a photosensor, a sonar, a lidar, a radar, a rangefinder, and a microphone.

NVIDIA Ex1002 Page 240 of 1081

6. The method of claim 1, wherein the first plurality of data readings represents at least one of a motor outflow, an optic flow, an inertial measurement unit measurement, a color measurement, a proximity measurement, a luminance measurement, a velocity measurement, an orientation measurement, an acceleration measurement, and a distance measurement.

7. The method of claim 1, wherein (A) comprises:

receiving a first set of data readings representing measurements by an inertial measurement unit (IMU) of the moving body;

receiving a second set of data readings representing commands issued to a motor of the moving body; and

receiving a third set of data readings representing an optic flow detected by an imaging sensor of the moving body.

8. The method of claim 7, wherein (C) comprises:

applying a first weight to the first set of data readings, a second weight to the second set of data readings, and a third weight to the third set of data readings;

increasing the first weight relative to the second weight and the third weight for data reading indicative of an increasing speed of the moving body;

increasing the third weight relative to the first weight and the second weight for data reading indicative of a decreasing speed of the moving body; and

varying the second weight from a value of zero for data readings indicative of no motion of the moving body to a value between the first weight and the third weight for data readings indicative of motion of the moving body.

9. The method of claim 8, further comprising:

(E) estimating the position and the orientation of the moving body with respect to the environment based on the second neural representation.

10. The method of claim 1, wherein (A) comprises:

receiving sensor data from an imaging sensor associated with the moving body;

deriving a first set of data readings from the sensor data, the first set of data readings representing at least one edge sensed by the imaging sensor;

NVIDIA Ex1002 Page 241 of 1081

deriving a second set of data readings from the sensor data, the second set of data readings representing an estimate of distance between the moving body and at least one object in the environment; and

deriving a third set of data readings from the sensor data, the third set of data readings representing motion of the moving body.

11. The method of claim 10, wherein (C) comprises:

applying a first weight to the first set of data readings, a second weight to the second set of data readings, and a third weight to the third set of data readings;

decreasing the first weight relative to the second weight and the third weight for data readings indicative of low contrast imagery sensed by the imaging sensor;

decreasing the second weight relative to the first weight and the third weight for data readings indicative of changes in depth below a predetermined threshold sensed by the imaging sensor; and

decreasing the third weight relative to the first weight and the second weight for data readings indicative of motion of the moving body.

12. The method of claim 1, wherein (B) comprises:

generating at least one first set of data cells representing respective data points in a measurement range of the at least one sensor, the respective data points representing a continuous portion of the environment;

defining a resolution of the first neural representation corresponding to the at least one first set of data cells based at least in part on a range of a corresponding set of data readings and on a number of data cells in the at least one first set of data cells;

determining a scale of the first neural representation corresponding to the at least one first set of data cells based at least in part on the range of the corresponding set of data readings; and

generating, with the processor, a set of graph connections among the at least one first set of data cells, the set of graph connections providing continuity between representations of the respective data points.

13. The method of claim 1, further comprising:

adjusting data readings received from the at least one sensor based at least in part on the plurality of aggregated data readings.

NVIDIA Ex1002 Page 242 of 1081

14. A system for navigating a robot through an environment, the system comprising:

an interface to receive first sensor data from a first sensor and second sensor data from a second sensor, the first sensor data and the second sensor data representing measurements of the environment with respect to the robot;

a memory, operably coupled to the interface, to store the first sensor data in a first neural representation of possible readings from the first sensor and to store the second sensor data in a second neural representation of possible readings from the second sensor; and

a path integration module, operably coupled to the memory, to:

(i) apply a first weight to the first sensor data in the first neural representation, the first weight representing a first measurement precision of the first sensor;

(ii) apply a second weight to the second sensor data in the second neural representation, the second weight representing a first measurement precision of the first sensor;

(iii) combine the first sensor data in the first neural representation with the second sensor data in the second neural representation so as to produce aggregated sensor data in an aggregated neural representation of the environment; and

(iv) estimate a position of the robot with respect to the environment based at least in part on the aggregated neural representation of the environment.

15. The system of claim 14, further comprising:

a map module, operably coupled to the path integration module, to generate a plurality of maps of the environment based on the position of the robot estimated by the path integration module.

16. The system of claim 15, further comprising:

a motivation and goal selection module, operably coupled to the path integration module, to implement a reward drive for the moving the robot and to generate a goal location for the robot based at least in part on the reward drive and the position of the robot estimated by the path integration module.

17. The system of claim 16, further comprising:

a path planning module, operably coupled to the path integration module, the map module, and the motivation and goal selection module, to generate a motor command for the robot based at least in part on the position of the robot estimated by the path integration

NVIDIA Ex1002 Page 243 of 1081

module, the plurality of maps of the environment generated by the map module, and the goal location generated by the motivation and goal selection module.

18. The system of claim 17, wherein the robot is configured to perform an action based on the motor command generated by path planning module.

19. The system of claim 16, wherein the reward drive comprises at least one of a curiosity drive, a consumable resource drive, and an external reward drive.

20. The system of claim 19, wherein the curiosity drive is configured to set the goal location for the robot in an unfamiliar area of the environment.

21. The system of claim 19, wherein the resource drive is configured to monitor a consumable resource of the robot and to instruct the robot to seek replenishment for the consumable resource when the consumable resource falls below a predetermined level.

22. The system of claim 19, wherein the external reward drive is configured to set the goal location in an area of the environment area where the robot had previously found an external reward.

23. The system of claim 19, wherein the goal location is based at least in part on a location of a reward object selected by the external reward dive, a location of a replenishment object selected by the resource drive, and a location selected by curiosity drive.

24. A method of providing an estimate of at least one of a position of a moving body and an orientation of the moving body, the method comprising:

(A) receiving a vestibular signal representative of the at least one of the orientation of the moving body and the position of the moving body;

(B) receiving a motor outflow signal representative of commanded motion of the moving body;

(C) receiving an optical flow signal representative of the at least of the orientation of the moving body and the position of the moving body with respect to an environment of the moving body;

(D) weighting the vestibular signal received in (A), the motor outflow signal received in (B), and the optical flow signal received in (C) based on a type of the motion of the moving body;

NVIDIA Ex1002 Page 244 of 1081

(E) combining the vestibular signal, motor outflow signal, and optical flow signal weighted in (D) to produce a localization signal; and

(F) generating, via at least one processor, the estimate of the at least one of the position of the moving body and the orientation of the moving body based on the localization signal produced in (E).

25. The method of claim 24, wherein (A) further comprises receiving the vestibular signal from at least one of a gyroscope and an accelerometer.

26. The method of claim 24, wherein (B) further comprises receiving the motor outflow signal from a controller operably coupled to a motor of the moving body.

27. The method of claim 24, wherein (C) further comprises:

(C1) receiving data from at least one of a camera, an infrared sensor, an ultrasonic sensor, a laser rangefinder, and a radio-frequency antenna; and

(C2) deriving the optical flow signal from the data received in (C1).

28. The method of claim 24, wherein (C) further comprises spatiotemporally filtering the optical flow signal.

29. The method of claim 24, wherein (D) further comprises at least one of:

(D1) determining rotation along at least one axis; and

(D2) determining translation along at least one axis.

30. The method of claim 24, wherein (D) further comprises selecting at least one weights based at least in part on at least one of a speed of the moving body, a velocity of the moving body, an acceleration of the moving body, and a trajectory of the moving body.

31. The method of claim 24, wherein the moving body comprises an autonomous vehicle.

32. A method for developing a robust noise-tolerant distributed data representation model of an environment based on data from at least one sensor, the method comprising:

(A) generating, with a processor, at least one set of data cells representing respective data points in a measurement range of the at least one sensor, the respective data points representing a continuous portion of the environment;

NVIDIA Ex1002 Page 245 of 1081

(B) defining, with the processor, a resolution of the neural representation corresponding to the at least one set of data cells based at least in part on a range of a corresponding set of data readings and on a number of data cells in the at least one set of data cells;

(C) determining, with the processor, a scale of the neural representation corresponding to the at least one set of data cells based at least in part on the range of the corresponding set of data readings; and

(D) generating, with the processor, a set of graph connections among the at least one set of data cells, the set of graph connections providing continuity between representations of the respective data points.

33. The method of claim 32, wherein the neural representation is first neural representation at a first scale and a first, and further comprising:

generating, with the processor, a second neural representation at a second scale and a second resolution according to steps (A) through (D);

combining, with the processor, outputs of the first neural representation with outputs of the second neural representation to produce a combined neural representation; and

estimating, with the processor, a position of a moving body associated with the at least one sensor based on the combined neural representation at a combined scale equal to least common multiple of the first scale and the second scale and a combined resolution equal to the smaller of the first resolution and the second resolution.

Figure 1A

Figure 1B

Figure 1C

Figure 1D

NVIDIA Ex1002 Page 247 of 1081

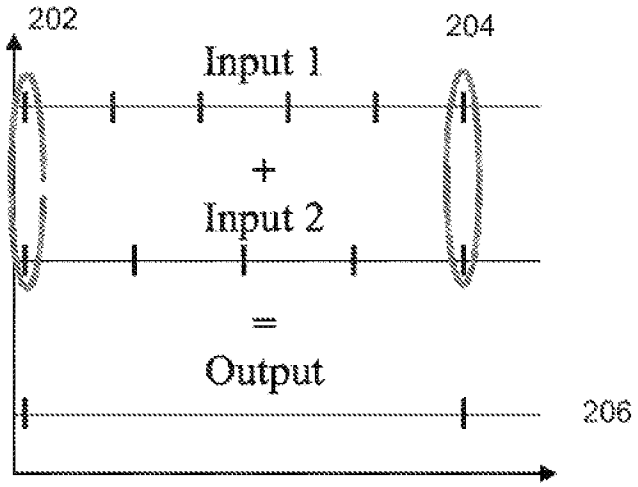


Figure 2.

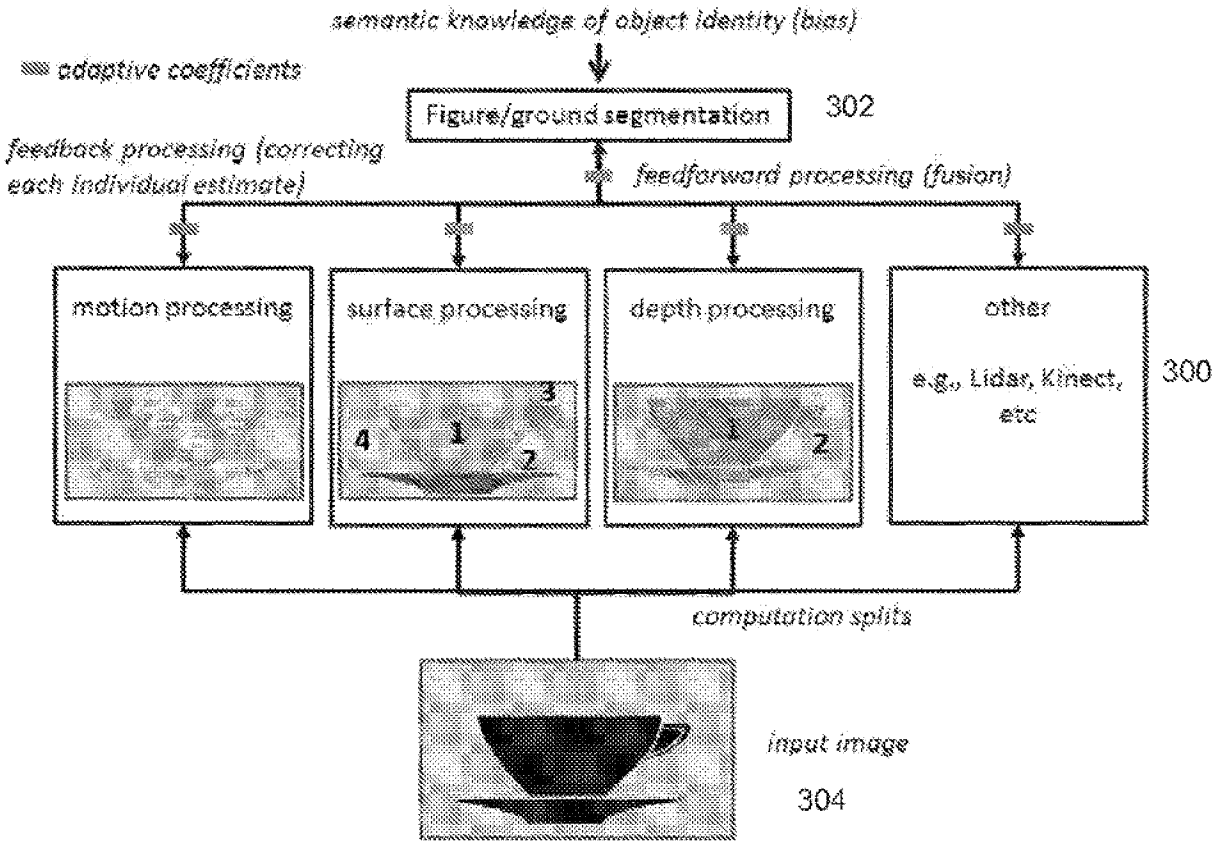

semantic knowledge of object identity (bias)
adaptive coefficients
Figure/ground segmentation
302
feedback processing (correcting each individual estimate)
feedforward processing (fusion)
motion processing
surface processing
depth processing
other
e.g., Lidar, Kinect, etc
300
computation splits
input image
304

Figure 3.

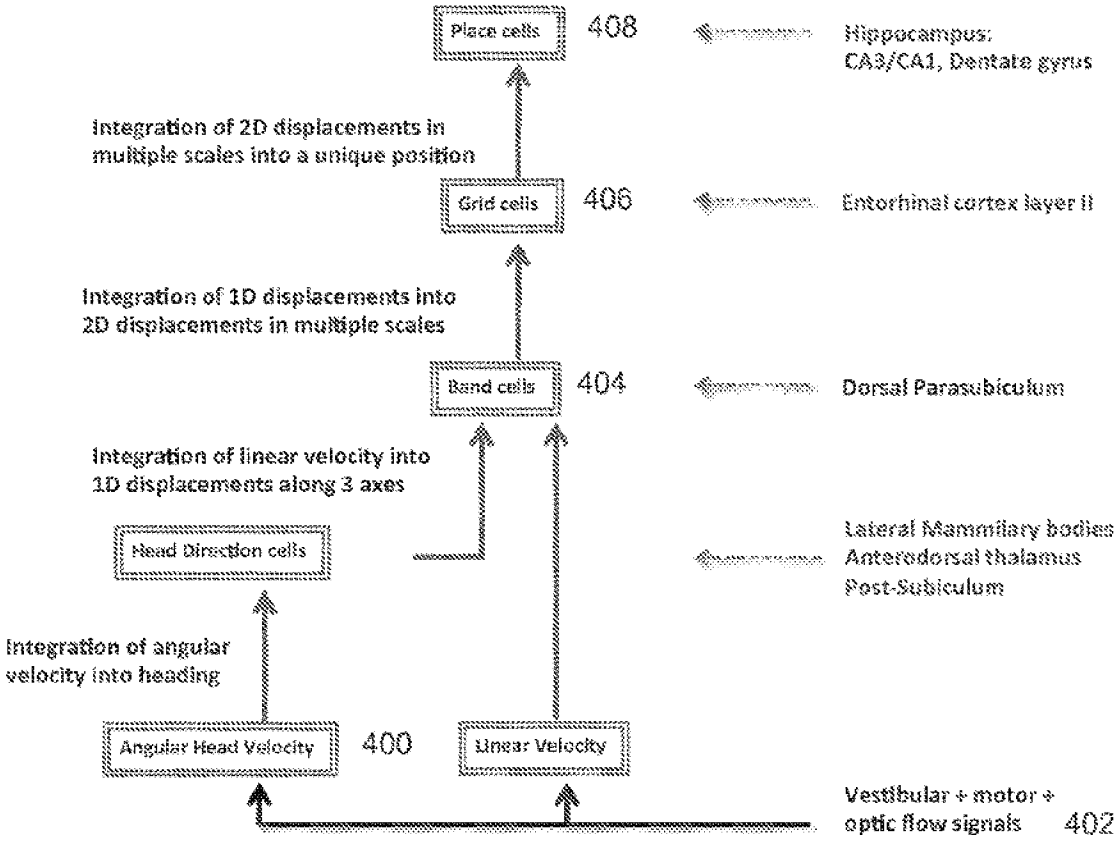

Place cells
408
Hippocampus:
CA3/CA1, Dentate gyrus
Integration of 2D displacements in multiple scales into a unique position
Grid cells
406
Entorhinal cortex layer II
Integration of 1D displacements into 2D displacements in multiple scales
Band cells
404
Dorsal Parasubiculum
Integration of linear velocity into 1D displacements along 3 axes
Head Direction cells
Lateral Mammillary bodies
Anterodorsal thalamus
Post-Subiculum
Integration of angular velocity into heading
Angular Head Velocity
400
Linear Velocity
Vestibular + motor + optic flow signals
402

Figure 4.

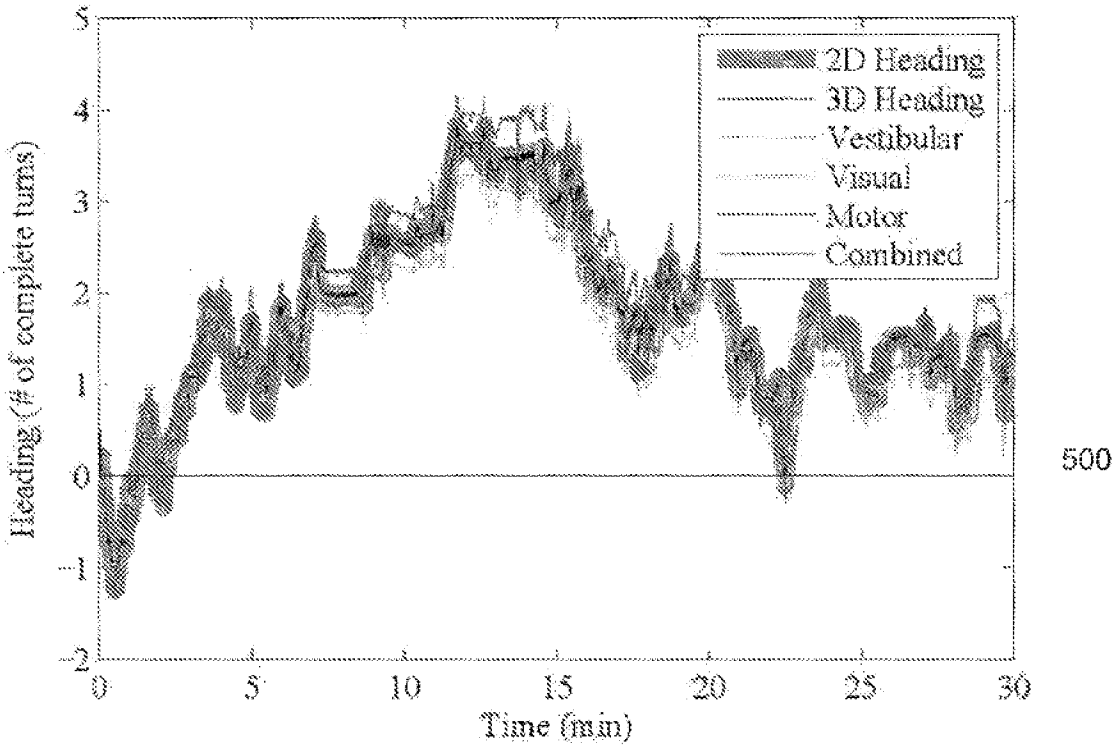

2D Heading
3D Heading
Vestibular
Visual
Motor
Combined
Heading (# of complete turns)
Time (min)
5
4
3
2
1
0
-1
-2
0
5
10
15
20
25
30
500

Figure 5.

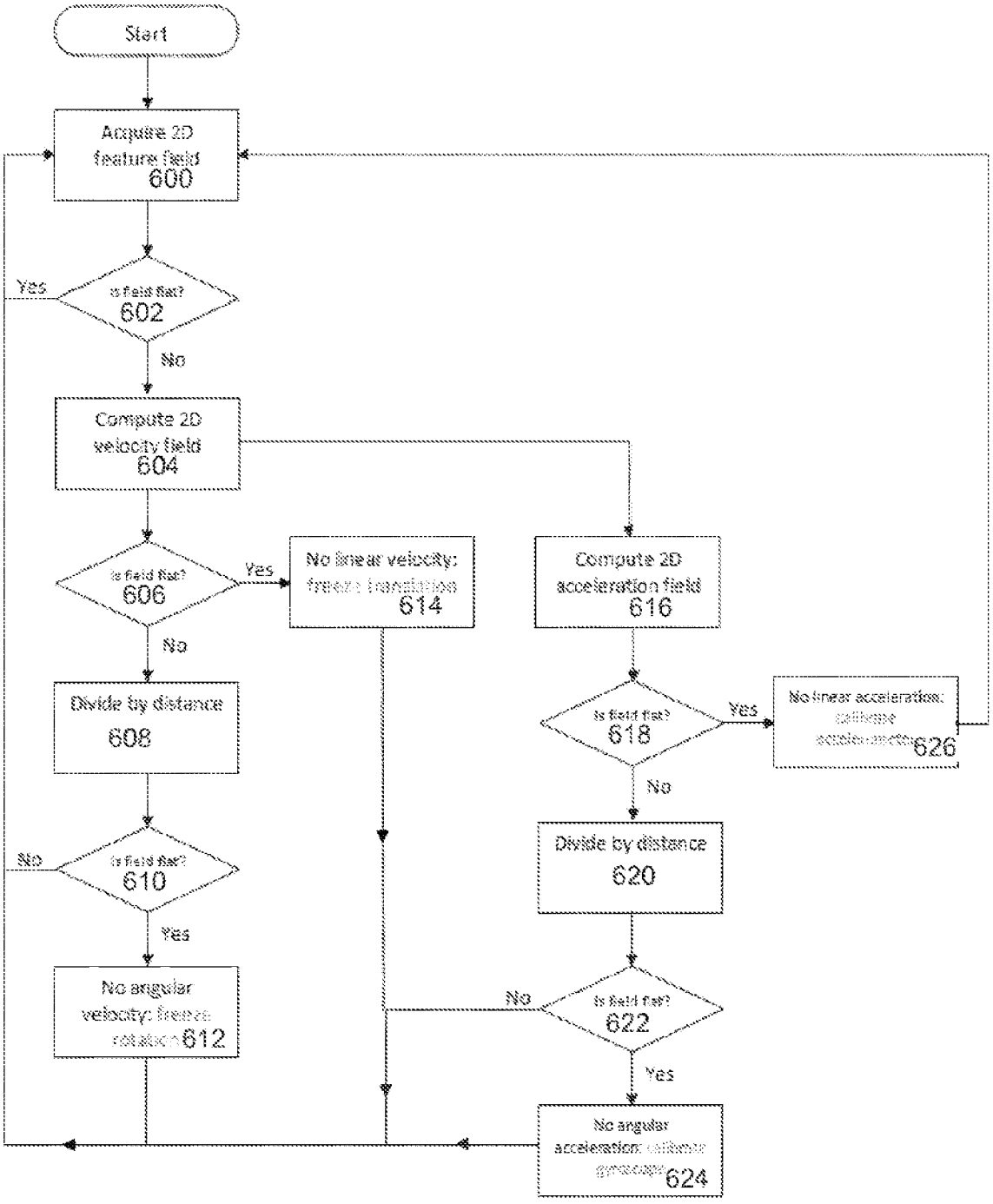

Start
Acquire 2D feature field 600
Is field flat? 602
Yes
No
Compute 2D velocity field 604
Is field flat? 606
Yes
No linear velocity: freeze translation 614
No
Divide by distance 608
Is field flat? 610
No
Yes
No angular velocity: freeze rotation 612
Compute 2D acceleration field 616
Is field flat? 618
Yes
No linear acceleration: calibrate acceleration 626
No
Divide by distance 620
Is field flat? 622
No
Yes
No angular acceleration: calibrate gyroscope 624

Figure 6.

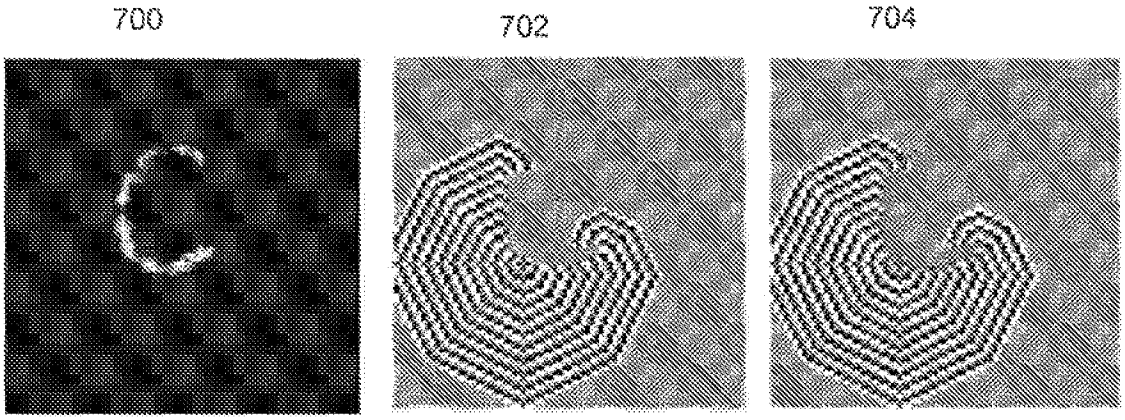


Figure 7.

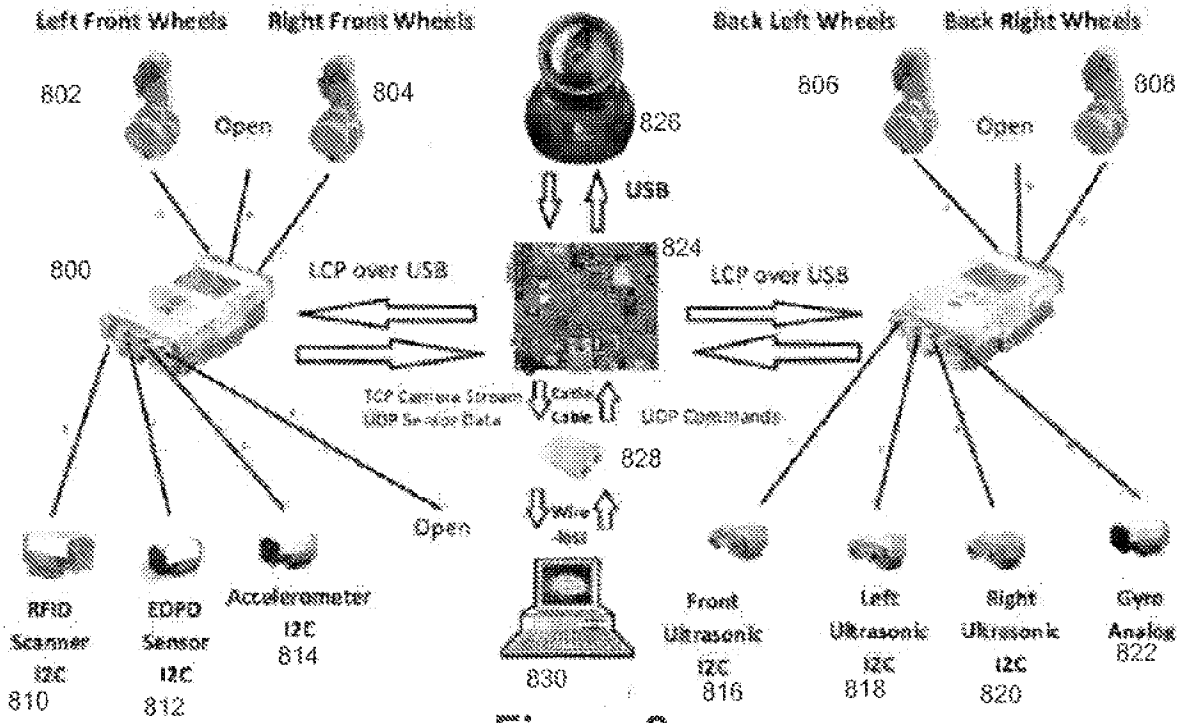

Left Front Wheels
Right Front Wheels
Back Left Wheels
Back Right Wheels
802
804
806
808
Open
Open
826
USB
824
800
LCP over USB
LCP over USB
828
Open
RFID Scanner I2C
810
EDPD Sensor I2C
812
Accelerometer I2C
814
830
Front Ultrasonic I2C
816
Left Ultrasonic I2C
818
Right Ultrasonic I2C
820
Gyro Analog
822

Figure 8.

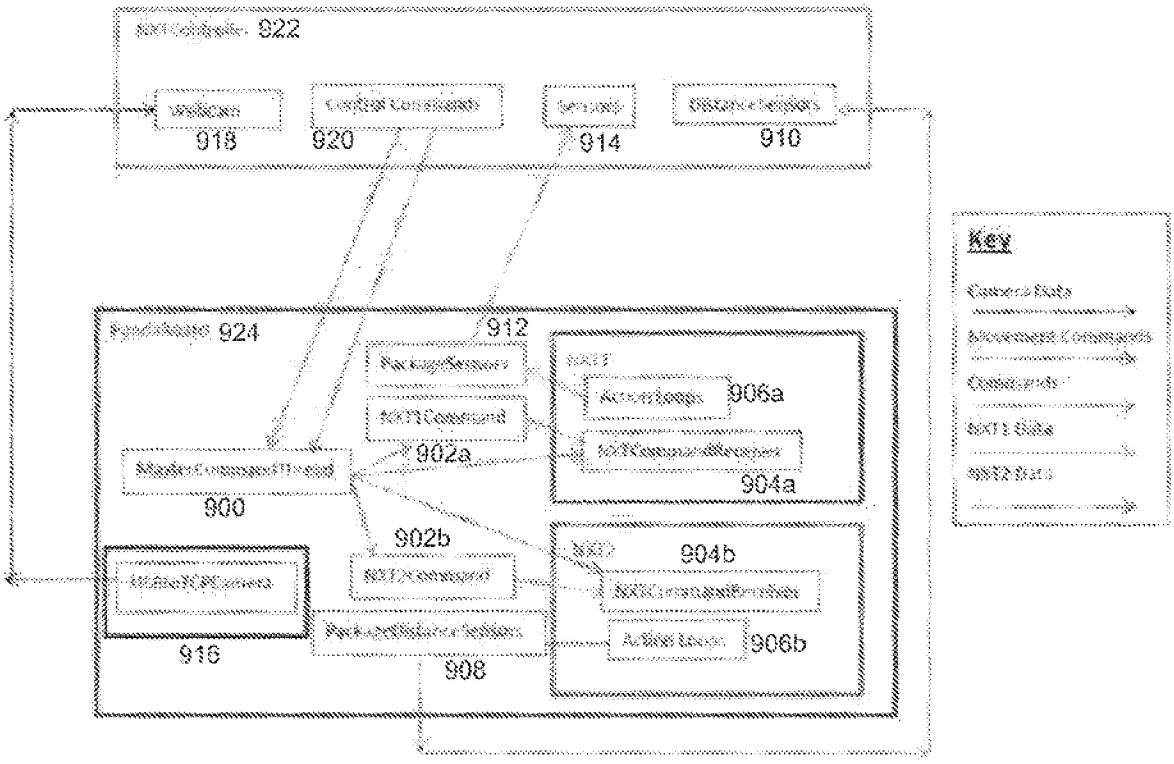

922
918
920
914
910
924
912
906a
902a
904a
900
902b
904b
906b
916
908
Key

Figure 9.

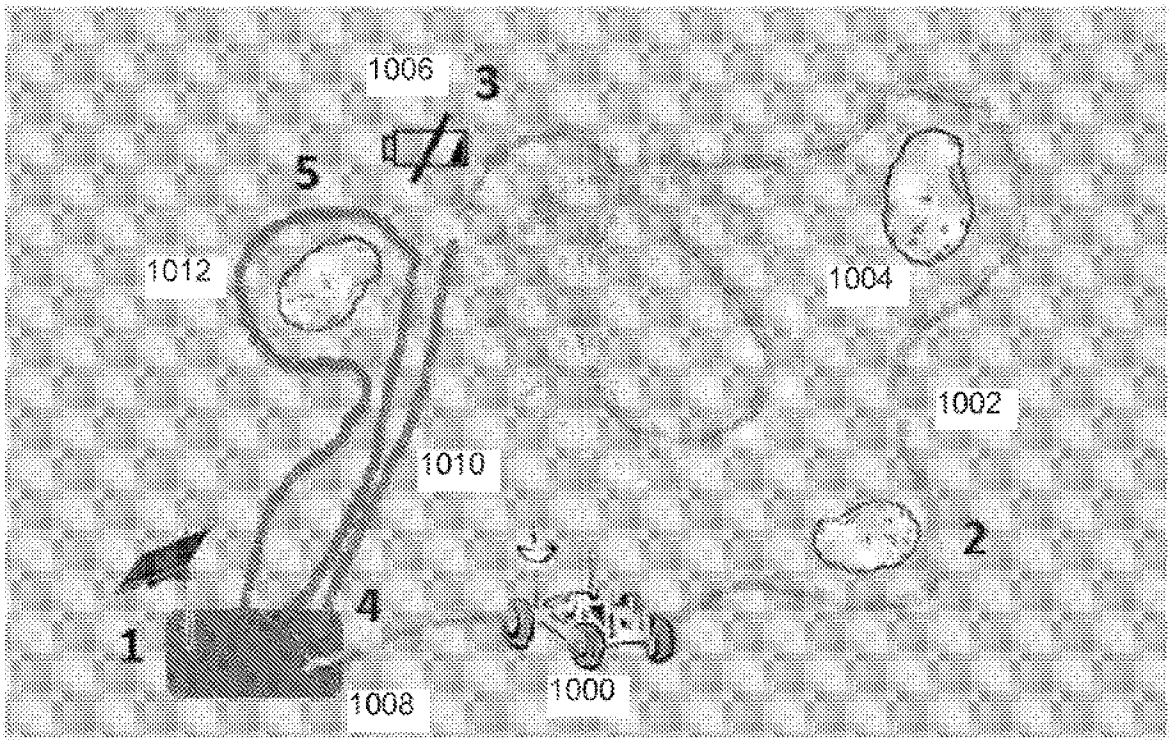


Figure 10.

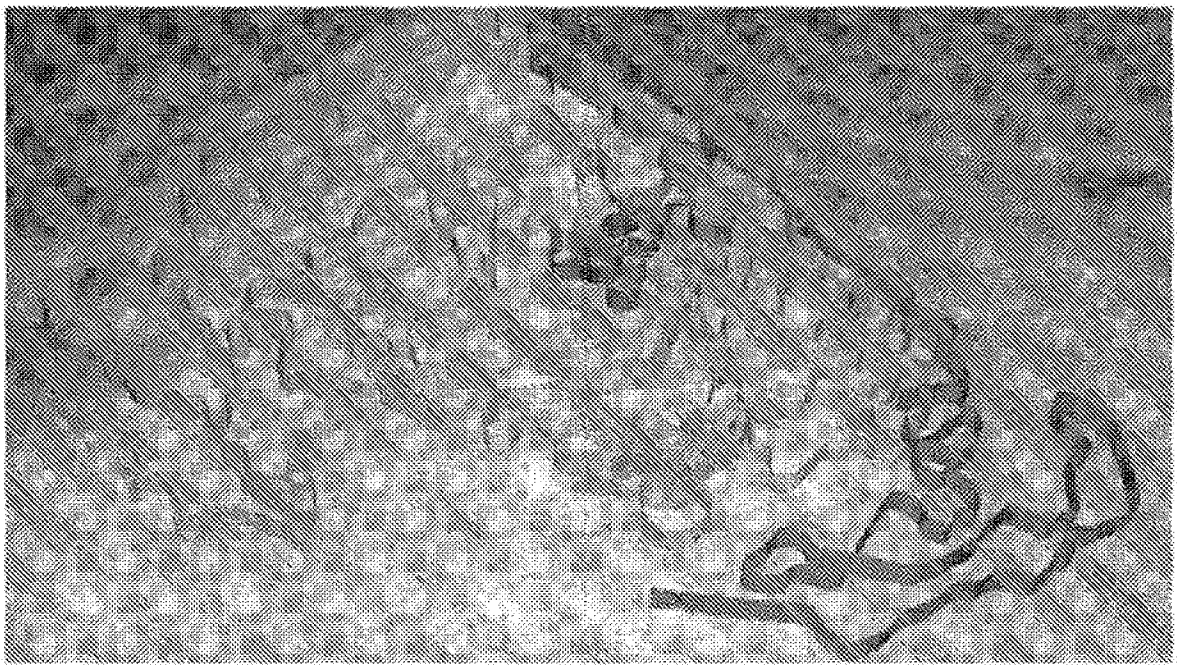


Figure 11.

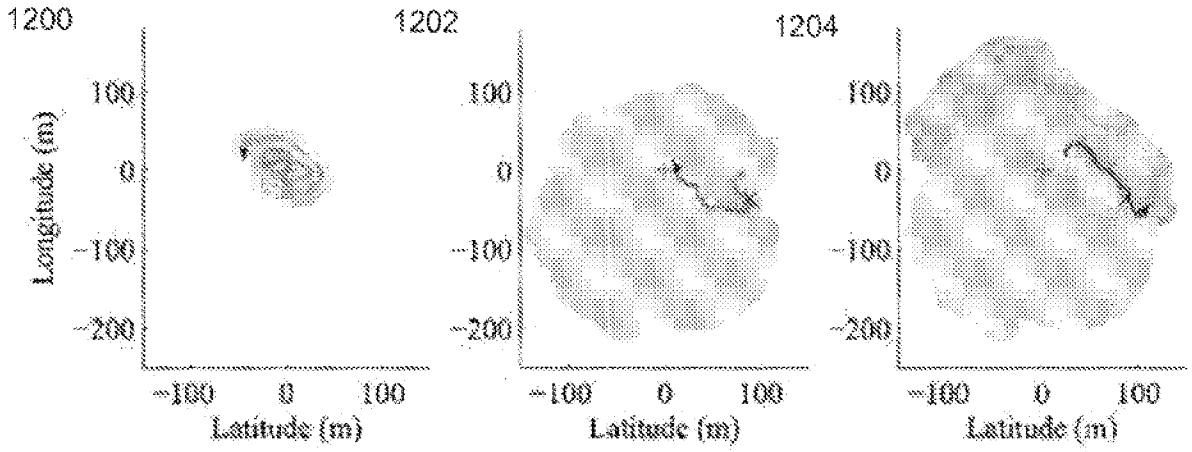


Figure 12.

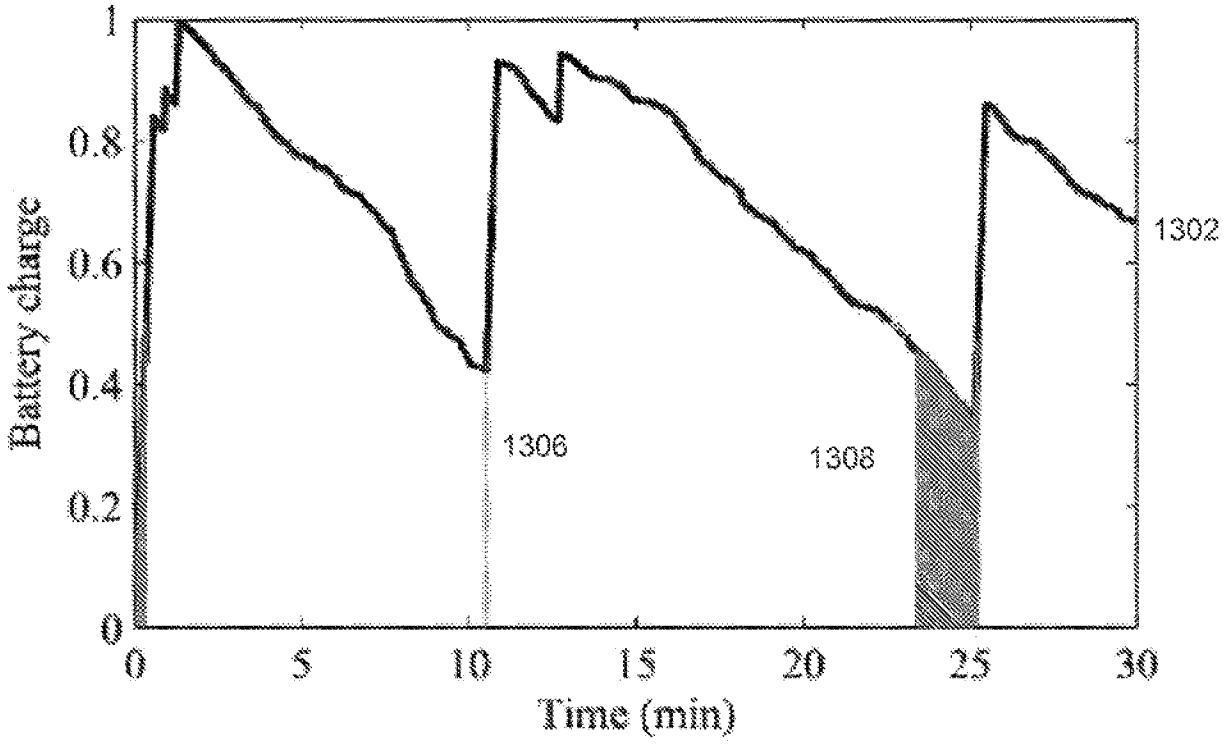

Battery charge
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20
25
30
Time (min)
1306
1308
1302

Figure 13.

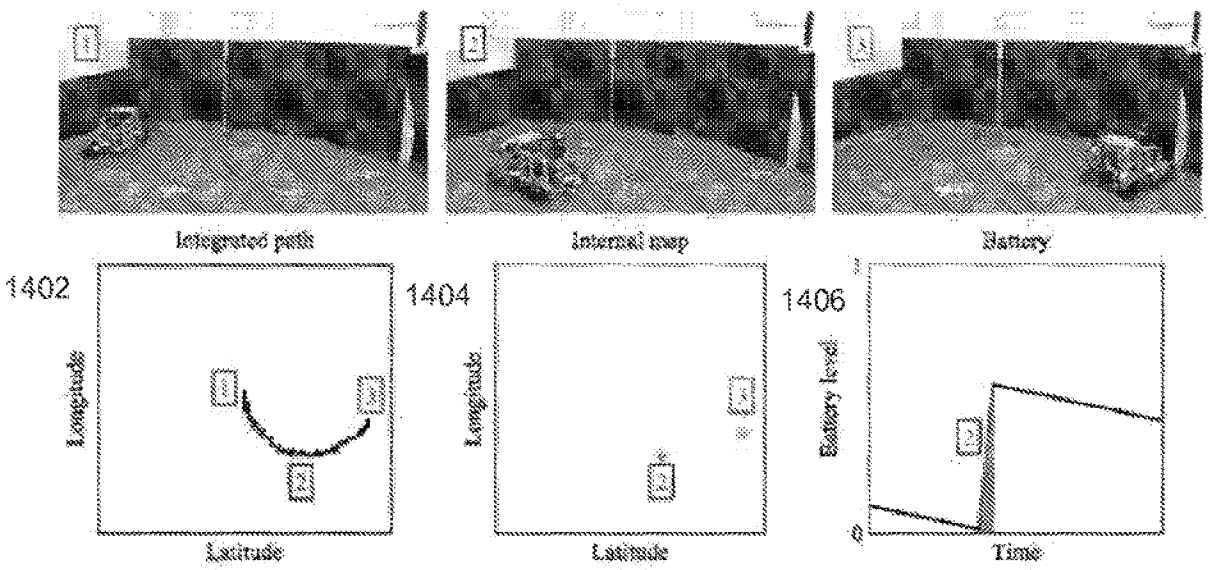

Integrated path
Internal map
Battery
1402
1404
1406
Longitude
Latitude
Longitude
Latitude
Battery level
Time

Figure 14.

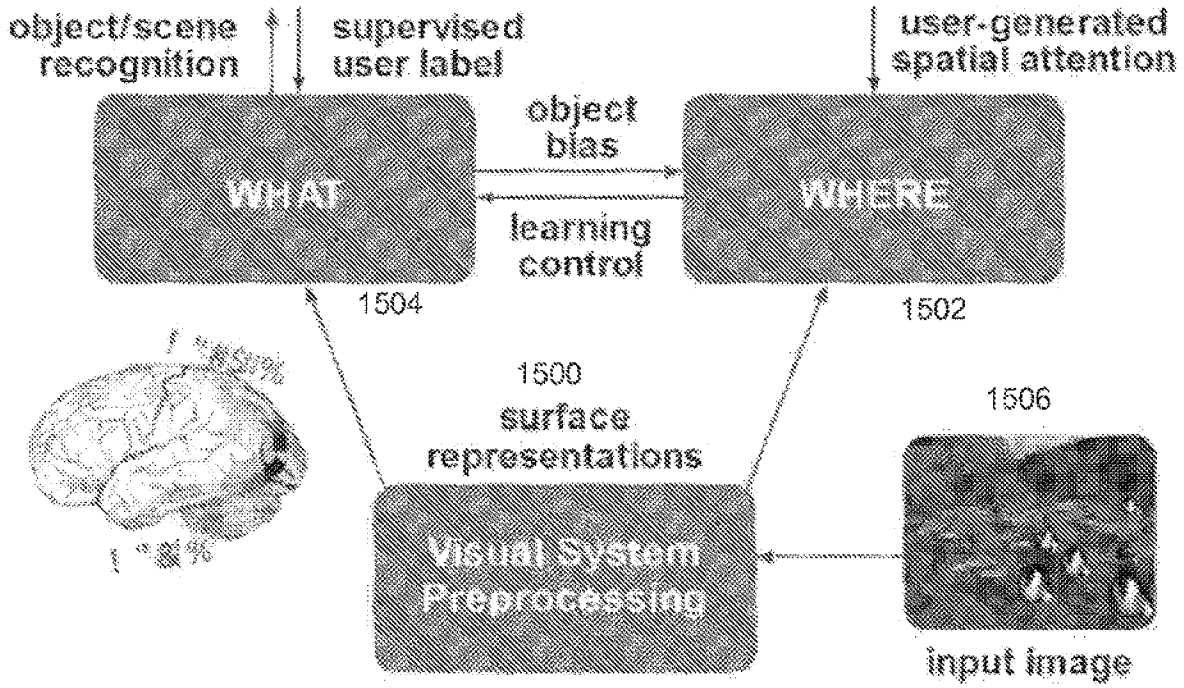
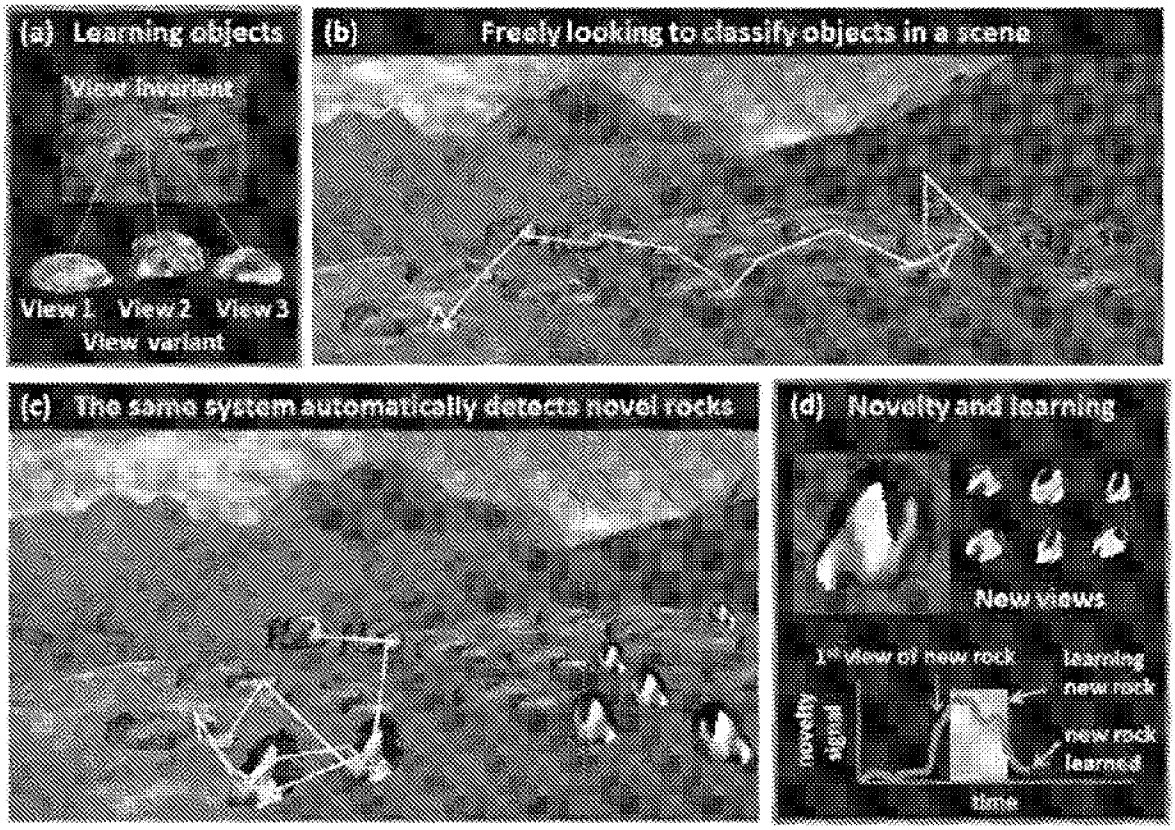

object/scene recognition
supervised user label
user-generated spatial attention
object bias
learning control
WHAT
WHERE
1504
1502
1500
surface representations
1506
Visual System Preprocessing
input image

Figure 15.

1600

1604



1606

Figure 16.

1602

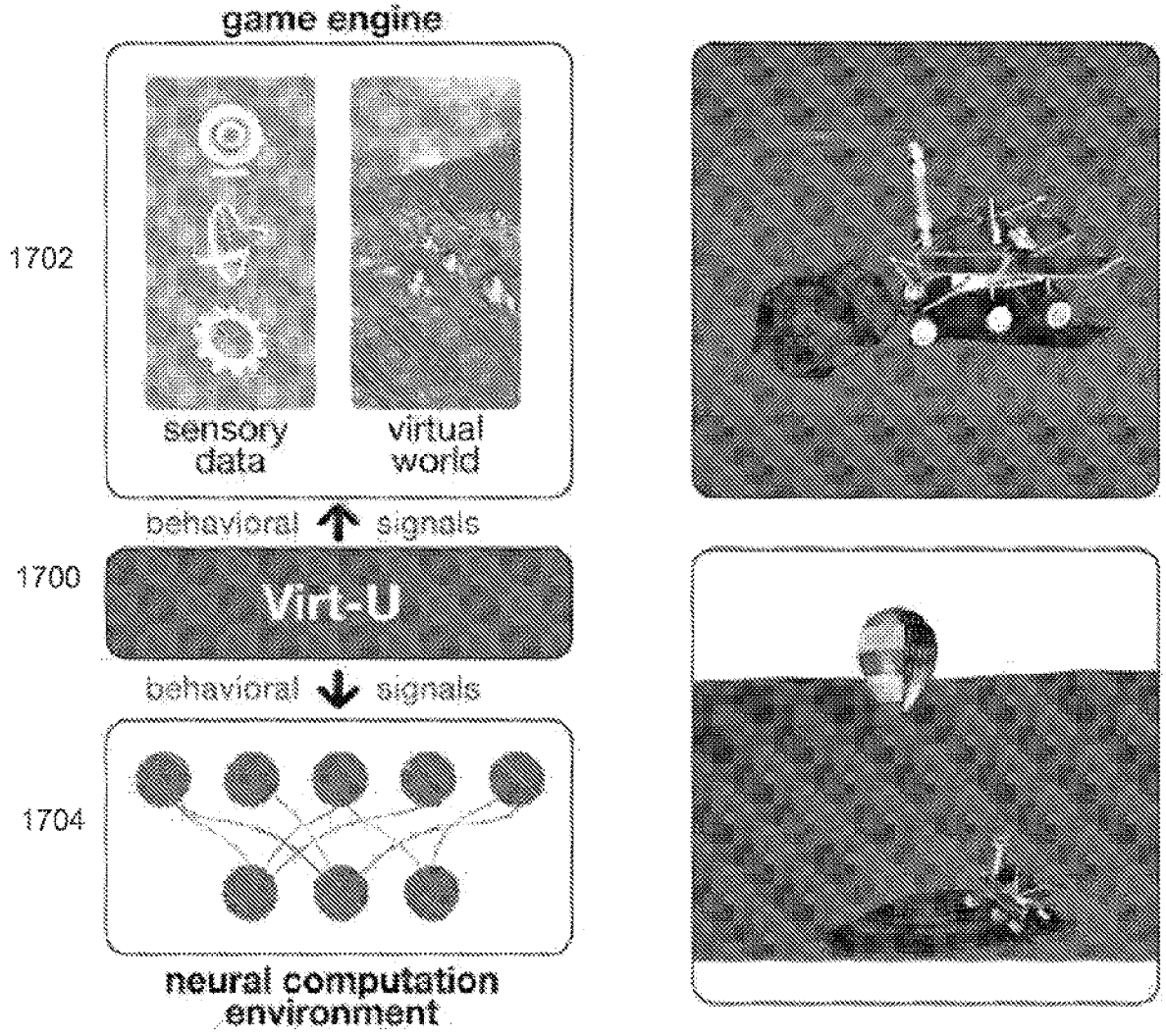

game engine
1702
sensory data
virtual world
behavioral signals
1700
Virt-U
behavioral signals
1704
neural computation environment

Figure 17.

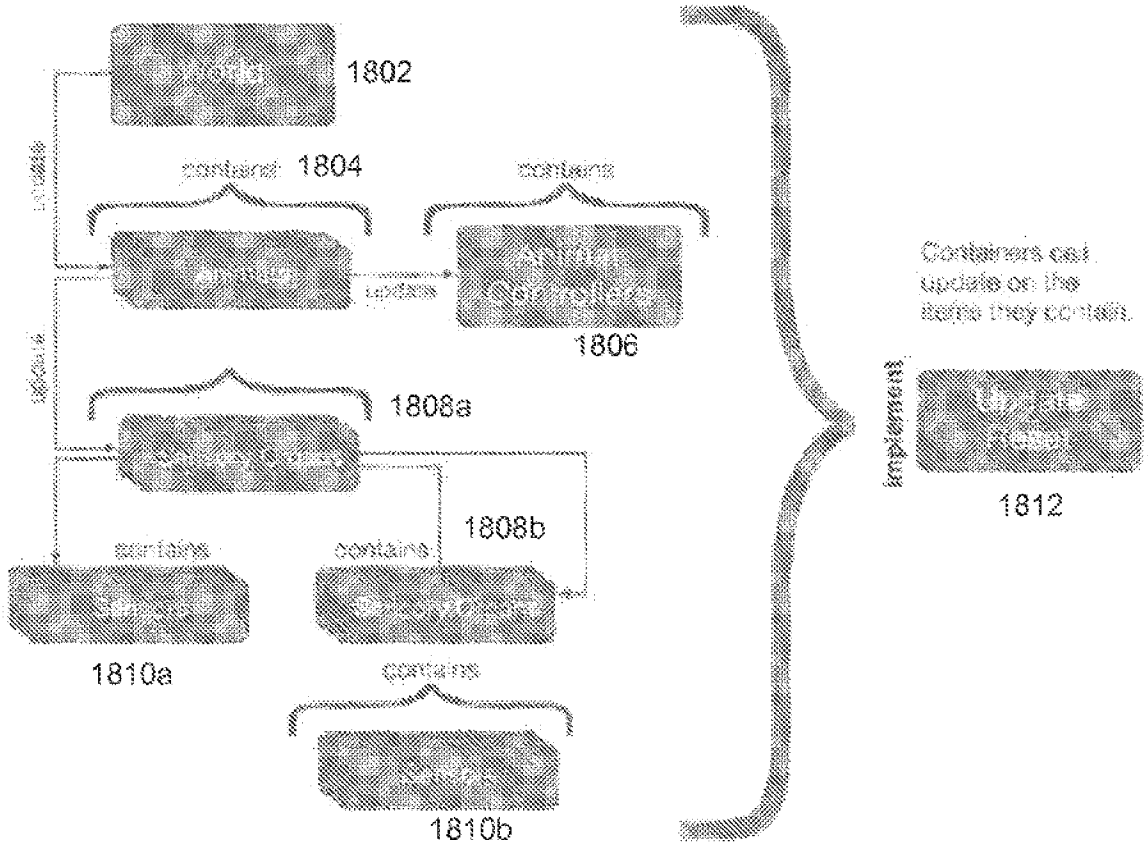

1802
contains
1804
contains
updates
1806
1808a
1808b
contains
contains
1810a
contains
1810b
implements
Containers call update on the items they contain.
1812
1800

Figure 18.

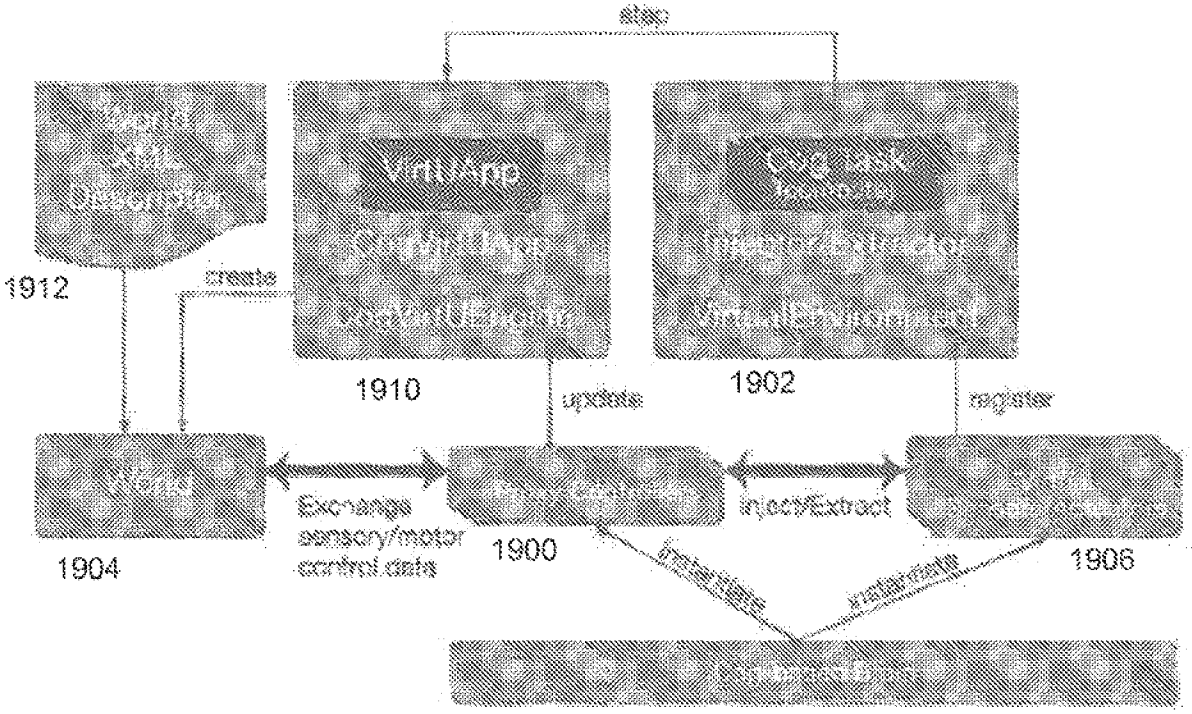

etc
Virtual Apps
create
update
register
Exchange sensory/motor control data
1900
Inject/Extract
1902
1904
1906
1908
1910
1912

Figure 19.

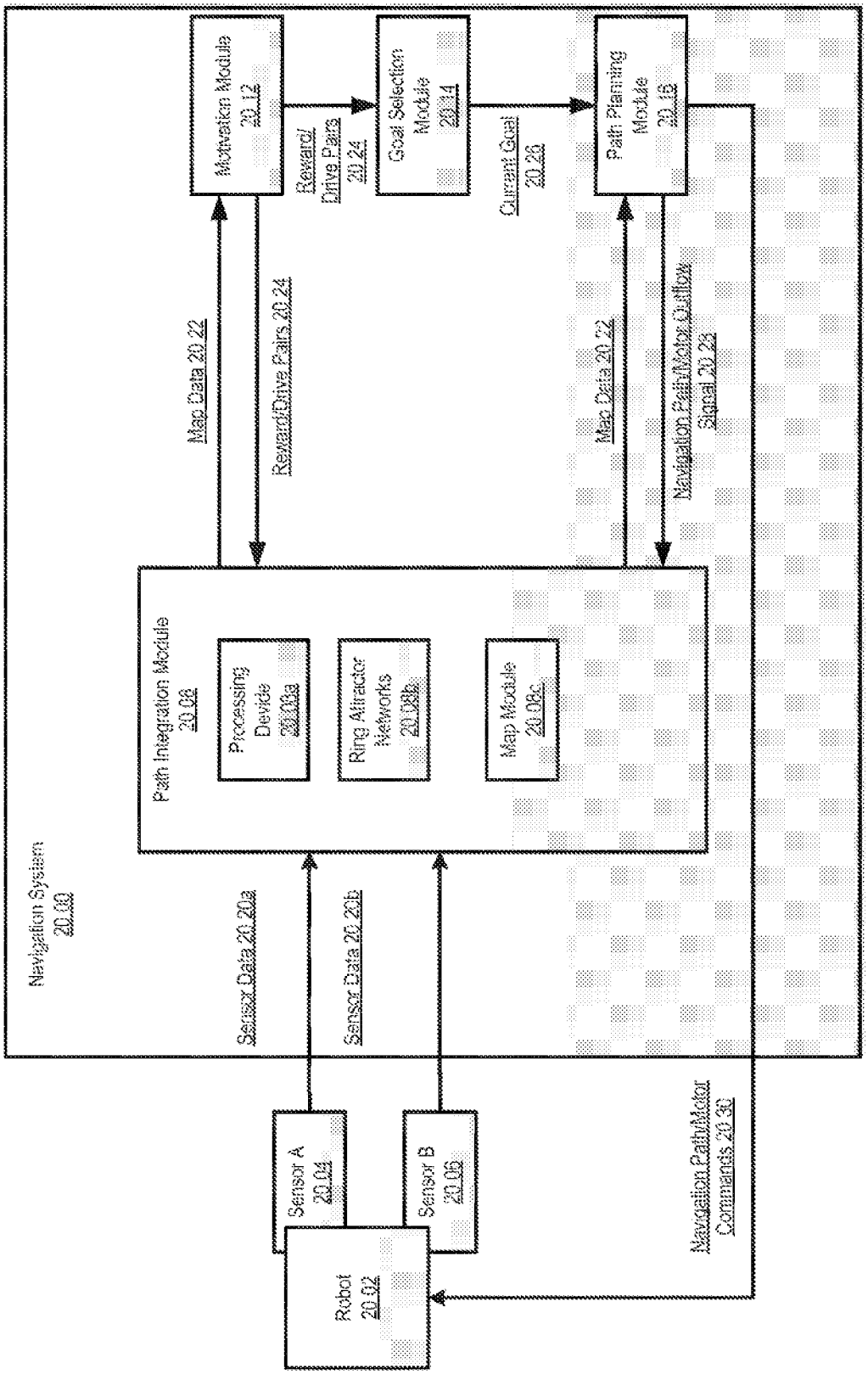

Navigation System
20.00
Motivation Module
20.12
Goal Selection Module
20.14
Path Planning Module
20.16
Reward/ Drive Pairs
20.24
Current Goal
20.26
Map Data 20.22
Reward/Drive Pairs 20.24
Map Data 20.22
Navigation Path/Motor Outflow Signal 20.28
Path Integration Module
20.08
Processing Device
20.08a
Ring Attractor Networks
20.08b
Map Module
20.08c
Sensor Data 20.20a
Sensor Data 20.20b
Sensor A
20.04
Sensor B
20.06
Robot
20.02
Navigation Path/Motor Commands 20.30

Figure 20

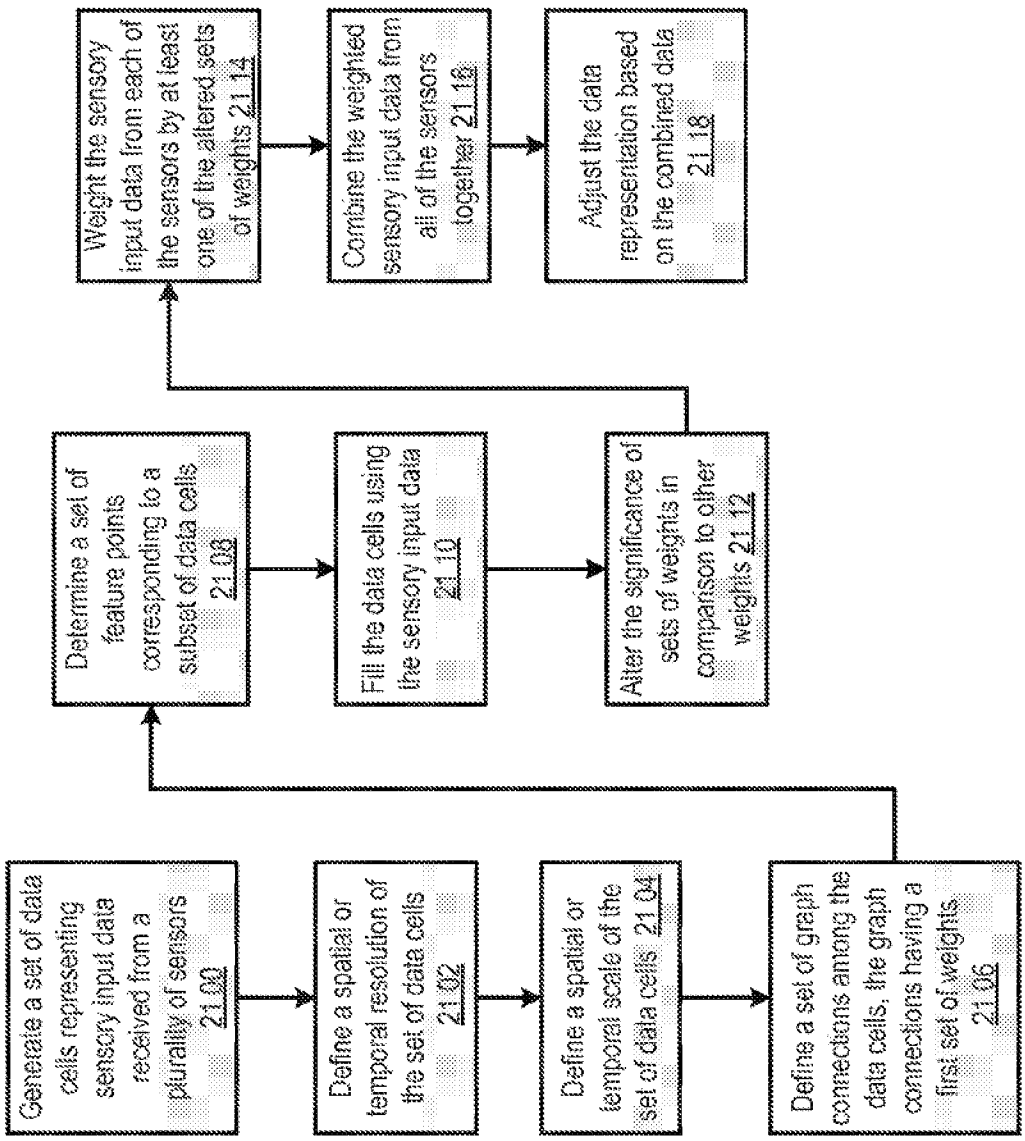

Generate a set of data cells representing sensory input data received from a plurality of sensors 21 00
Define a spatial or temporal resolution of the set of data cells 21 02
Define a spatial or temporal scale of the set of data cells 21 04
Define a set of graph connections among the data cells, the graph connections having a first set of weights 21 06
Determine a set of feature points corresponding to a subset of data cells 21 08
Fill the data cells using the sensory input data 21 10
Alter the significance of sets of weights in comparison to other weights 21 12
Weight the sensory input data from each of the sensors by at least one of the altered sets of weights 21 14
Combine the weighted sensory input data from all of the sensors together 21 16
Adjust the data representation based on the combined data 21 18

Figure 21

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 December 2014 (24.12.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/204615 A2

(51) International Patent Classification:
G06F 9/315 (2006.01)

(21) International Application Number:
PCT/US2014/039162

(22) International Filing Date:
22 May 2014 (22.05.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/826,380 22 May 2013 (22.05.2013) US

(71) Applicant: **NEURALA, INC.** [US/US]; The Photonics Center, 8 St. Mary's Street, Suite 613, Boston, Massachusetts 02215 (US).

(72) Inventors: **GORSHECHNIKOV, Anatoly**; 15 Jefferson Street, Newton, Massachusetts 02458 (US). **VERSACE, Massimiliano**; 846 East 3rd Street, Boston, Massachusetts 02127 (US). **VERSACE, Heather Ames**; 846 East 3rd Street, Boston, Massachusetts 02127 (US). **LIVITZ, Gennady**; 61 Brookside Avenue, Belmont, Massachusetts 02478 (US).

(74) Agents: **TEJA, Joseph** et al.; Cooley LLP, 1299 Pennsylvania Ave., Suite 700, Washington, District of Columbia 20004 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHODS AND APPARATUS FOR ITERATIVE NONSPECIFIC DISTRIBUTED RUNTIME ARCHITECTURE AND ITS APPLICATION TO CLOUD INTELLIGENCE

(57) Abstract: The system and methods disclosed herein include a runtime architecture that takes a non-specific set of systems of differential equations, distributes them across the network, and iteratively integrates them through time with a possibility to output the results on every iteration. Embodiments of the disclosed system may be used for neural computation or any other suitable application. Embodiments can be used as a standalone engine or as part of another computational system for massively parallel numerical integration of a data-driven dynamical system.

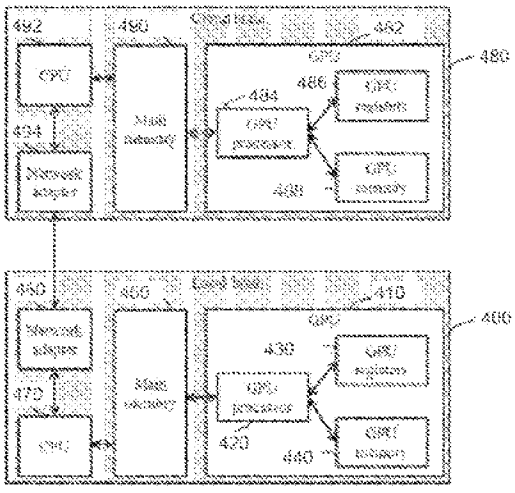


Figure 4

WO 2014/204615 A2

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

Methods and Apparatus for Iterative Nonspecific Distributed Runtime Architecture and its Application to Cloud Intelligence

CROSS-REFERENCE TO RELATED PATENT APPLICATION

[0001] This application claims priority, under 35 U.S.C. § 119(e), from U.S. Application No. 61/826,380, filed May 22, 2013, and entitled "Iterative Nonspecific Distributed Runtime Architecture," which application is hereby incorporated herein by reference in its entirety.

GOVERNMENT SUPPORT

[0002] This invention was made with government support under Contract No. FA8750-12-C-0123 awarded by the U.S. Air Force. The government has certain rights in the invention.

BACKGROUND

[0003] Current computer hardware central processor architectures and often even graphic processor enabled workstations are often unable to achieve the highly parallel requirements needed to perform computationally expensive massive data-driven algorithms, especially those that are based on dynamic systems approach.

SUMMARY

[0004] The system and methods described herein are based on a network of processing nodes that can be individual heterogeneous workstations or a cluster of homogeneous computer boards. An example of the system enables a runtime architecture based on this hardware to load a nonspecific set of systems of differential equations, distribute them across network nodes according to hardware capabilities of each node, and iteratively integrate them through time with a possibility to output the results on every iteration. An exemplary system may be used for a variety of applications, including but not limited to neural computations. For instance, an exemplary system can be used as a standalone computational engine engaged on a hardware cluster directly or as a part of other computational software that at some point uses massively parallel numerical integration of data driven dynamical systems on a network of remote computational hardware. Furthermore, a resulting computational engine can serve as a remote (e.g., cloud) back end that can engage remote hardware resources to support local autonomous hardware devices such as robots, automatic household devices, mobile navigators, and other devices.

NVIDIA Ex1002 Page 270 of 1081

[0005] In some implementations, a method for processing sensory data in parallel may comprise receiving, at a master node, a request to process incoming sensor data, receiving, from the master node, a selection of a set of slave nodes, the set of slave nodes selected from a set of nodes capable of processing the incoming sensor data, and establishing a connection between the master node and the set of slave nodes. For at least one slave node in the set of slave nodes, the implementation may further include determining a dynamic system model for a portion of the incoming sensor data received by the at least one slave node, distributing, from the master node to the at least one slave node, a representation of the dynamic system model and a representation of corresponding graph connections between slave nodes in the set of slave nodes, instantiating at least one local data structure for processing the incoming sensor data based at least in part upon the representation of the dynamic system model distributed earlier, connecting the at least one slave node to other slave nodes in the set of slave nodes according to the representation of the graph connections distributed earlier, receiving a signal from the master node to process the incoming sensor data; and iteratively processing the incoming sensor data using the at least one local data structure instantiated previously.

[0006] In some implementations, the process may also comprise selecting the set of slave nodes automatically, or selecting the set of slave nodes according to user input. In some implementations, distributing the representation of the dynamic system model may further comprise assigning at least one system of equations corresponding to the dynamical system model to the at least one slave node based on at least one of computational power of the at least one slave node, computational complexity of the at least one system of equations, and communication between parts of a dynamical system represented by the dynamical system model. Distributing the representation of the dynamic system model may also comprise assigning, by the at least one slave node, a set of computational threads for processing at least a portion of the incoming sensor data.

[0007] In some implementations, connecting the at least one slave node to other slave nodes may further comprise determining an operating mode of the master node, and connecting the master node to the at least one slave node based on the operating mode of the master node. In some implementations, this may further comprise connecting the master node to the at least one slave node via a network connection if the master node is configured to operate in a networked mode, and compiling the slave node into the master node if the master node is configured to operate in a standalone mode. In some implementations the master node is

NVIDIA Ex1002 Page 271 of 1081

configured to operate in a networked mode or a standalone mode based on available resources, and is configured to operate in the networked mode if the master node is able to connect to a network and if there is at least one slave node connected to the network.

[0008] In some implementations, processing the data may further comprise storing results of the iterative processing in a data queue in the at least one slave node for communication with the other slave nodes in the set of slave nodes, and storing results of the iterative processing in a storage medium for off-line analysis.

[0009] An exemplary system for processing and forwarding slave node data may comprise a first slave node operatively coupled to an outgoing axonal queue associated with a system of equations processed on the first slave node, a shadow axonal queue corresponding to respective outgoing axonal queues of other slave nodes configured to provide input data to the first slave node, a set of synaptic queues corresponding to the system of equations processed on the first slave node, the set of synaptic queues configured to preprocess the input data received from the shadow axonal queue, and a processing device, operably coupled to the set of synaptic queues, to apply the system of equations to the input data preprocessed by the synaptic queue so as to generate at least one output data value. The system may further comprise a local outgoing axonal queue, operatively coupled to the processing device, to receive the at least one output data value from the processing device and to provide the at least one output data value to a second shadow axonal queue in at least one of the first slave node or a second slave node. The shadow axonal queue may be configured to receive at least one data value from a corresponding outgoing axonal queue operatively coupled to a slave node.

[0010] The outgoing axonal queues and shadow axonal queues can be implemented with a standard template library (STL) class called multimap that holds key-value pairs. The key in the key-value pair may be event production time and the value in the key-value pair can include for example receiver address and output value. STL multimap does automatic sorting by key, so this queue may have output events sorted by their production times. The synaptic queue can also be implemented with a standard template library (STL) class called multimap that holds key-value pairs, except that the receiver address is no longer needed, the timing represents delivery time rather than production time, and the value in the key-value pair is adjusted according to the parameters of the synapse.

NVIDIA Ex1002 Page 272 of 1081

[0011] An exemplary system for aggregating and processing sensor data from a robot comprising a local processor that implements a local brain may comprise at least one slave node, located on a computer network, to implement a cloud brain for receiving and processing the sensor data from the robot, the first processor including a main memory module to store the sensor data from the robot, and a communications device, operably coupled to the main memory module, to transmit at least a portion of the sensor data to the local brain implemented in the local processor of the robot and to receive, from the local brain, at least one data value based at least in part on the at least a portion of the sensor data and on a current state of the local brain.

[0012] The main memory module may store a representation of at least one of an outgoing axonal queue associated with a system of equations processed on a first slave node, a shadow axonal queue corresponding to respective outgoing axonal queue of a second slave node configured to provide input data to the first slave node, a synaptic queues corresponding to the system of equations processed on the first slave node, the set of synaptic queues configured to preprocess the input data received from the shadow axonal queue, and a data structure for processing the sensor data from the robot. The communications device is operably coupled to another local brain implemented in another robot and configured to transmit the at least one data value to the other local brain. The communications device may be further configured to transmit data from a dedicated memory block in the main memory module to the other local brain and to receive, from the other local brain, at least one other data value based at least in part on the data from the dedicated memory block.

[0013] It should be appreciated that all combinations of the foregoing concepts and additional concepts discussed in greater detail below (provided such concepts are not mutually inconsistent) are contemplated as being part of the inventive subject matter disclosed herein. In particular, all combinations of claimed subject matter appearing at the end of this disclosure are contemplated as being part of the inventive subject matter disclosed herein. It should also be appreciated that terminology explicitly employed herein that also may appear in any disclosure incorporated by reference should be accorded a meaning most consistent with the particular concepts disclosed herein.

NVIDIA Ex1002 Page 273 of 1081

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The skilled artisan will understand that the drawings primarily are for illustrative purposes and are not intended to limit the scope of the inventive subject matter described herein. The drawings are not necessarily to scale; in some instances, various aspects of the inventive subject matter disclosed herein may be shown exaggerated or enlarged in the drawings to facilitate an understanding of different features. In the drawings, like reference characters generally refer to like features (e.g., functionally similar and/or structurally similar elements).

[0015] Figures 1A and 1B are diagrams of a networked system and a standalone system each implementing Iterative Nonspecific Distributed Runtime Architectures, respectively, according to illustrative implementations.

[0016] Figure 2 is a flow chart of the control flow through the systems of Figures 1A and 1B, according to one illustrative implementation.

[0017] Figures 3A and 3B illustrate data flow on the slave hardware nodes of system of Figure 1A and standalone system of Figure 1B, according to illustrative implementations. Figure 3A shows data flow during system and model setup. Figure 3B shows the data flow during the computation.

[0018] Figure 3C is a flow chart of a method for event processing, according to one illustrative implementation.

[0019] Figure 3D is a flow chart of event communications between hardware nodes on the network, according to one illustrative implementation.

[0020] Figure 4 is a diagram of an illustrative implementation of a cloud server (brain) and a local robot, according to one illustrative implementation.

[0021] Figure 5 is a diagram of an illustrative implementation of a cloud server (brain) and a plurality of local robots, according to one illustrative implementation.

[0022] Figure 6 is a diagram of an illustrative implementation of a cloud server (brain) with separation of processors and memory dedicated to a plurality of robots, according to one illustrative implementation.

NVIDIA Ex1002 Page 274 of 1081

[0023] Figure 7 is a diagram of an illustrative implementation of transferring knowledge to a first robot from a portion of a cloud server (brain) dedicated to a second robot, according to one illustrative implementation.

[0024] Figure 8 is a diagram of an illustrative implementation of sharing knowledge on a cloud server (brain) without copying network structure.

DETAILED DESCRIPTION

[0025] The system and methods described herein relate to a hardware computational engine consisting of two or more computational nodes such as workstations or cluster components connected through a wired or wireless network. In some implementations, the engine underlies and subserves neural simulations and/or other simulations. The system employs network communication management, thread management, event based communication, over the network data exchange, and data structures used to facilitate the functionality herein.

[0026] In some implementations, the system includes a computer readable memory onto which processor executable instructions are written. The computer readable memory can store a plurality of libraries. For example, the system on a slave computational node may include: a pthreads library that is used to manage multithreaded control flow; a standard template library (STL) that may be used for structuring the data; and a CUDA or OpenCL library to control execution on graphic processors. In some implementations, the graphics processor unit (GPU) libraries may not be required. The system on the master node may include additional libraries for graphic user interface and data visualization and/or other additional libraries and/or like information.

[0027] Figure 1A is a system diagram of a networked implementation 100 of an Iterative Nonspecific Distributed Runtime Architecture (INDRA). The networked implementation 100 includes a single master node 102 receiving commands through an application programming interface (API) and controlling an arbitrary number of slave nodes 106a-b. Figure 1B shows a standalone version 108 of an INDRA receiving commands through the same API (master part) 110 and executing the commands through a simplified version of a slave node 112 created and connected to locally by the master node (e.g., the slave node 112 may be directly merged into the master node).

[0028] As illustrated in Figure 1A, the structure of the system 100 comprises a single master node 102 and an arbitrary number of slave nodes 106a, 106b, ... (collectively, slave

NVIDIA Ex1002 Page 275 of 1081

nodes 106). The master node 102 is a computing hardware component where a user or an external application can instantiate the engine control structures (those engine control structures including but not limited to a parser that converts user provided description of a dynamical system in machine readable data structures, a network communication manager that establishes connections with slaves, and a command and synchronization module that passes commands (e.g. run, stop, reset) and synchronization signals from master to the slaves). It can be implemented as a user controlled workstation, portable electronic device, or as an autonomous electronic device such as robot. The engine control structures may initiate engine execution on remote computer systems serving as slave nodes 106 to the engine, gather the information about available hardware resources for each slave (CPU cores, GPUs, memory), and distribute the dynamical systems, which may take the form of neuronal populations (or systems of differential or algebraic equations) and corresponding synapses (event-based dependencies between these equations) across the slave nodes 106.

[0029] As used herein, a dynamical system is a set of systems of differential and or algebraic equations that describe some natural process through time. A neural model is a subset of a dynamical system where equations represent neuronal dynamics. A neuronal population is a part of a neural model that comprises at least one neuronal element and for all neuronal elements the governing system of equations is identical up to a parameter choice. A synapse is a dependency between two neuronal elements that can be in the same or different populations such that it connects a source neuronal element with the target neuronal element and passes output events from the source to the target according to some rule defined by the user.

[0030] The above tasks correspond to the System Setup and Model Initialization phases of the example flow chart in Figure 2 and example data exchange diagram in Figure 3A. During these stages the slave nodes 106 receive network communication from the master node control structures and instantiate local data structures, allocate hardware memory, query processing devices, and otherwise prepare to perform the computation on the local hardware.

[0031] The dynamical system may comprise a set of populations, where each population is a collection of identical systems of equations, and where each element in a population is a computational element.

[0032] Each computational element may have output variables that affect the rest of the system. For example, in the case of a neural model, the output of a neuron may be those

NVIDIA Ex1002 Page 276 of 1081

output variables. A computational element also may have several internal variables that are used to compute output variables, but are not exposed to the rest of the system, not even to other elements of the same population, typically. Each of these variables is represented as a CPU or GPU memory chunk. An exemplary difference between output variables and internal variables is their access: output variables may be made available to other computational elements either directly or through generation of output events, while internal variables may be hidden from other computational elements.

[0033] An example implementation of a computational element data structure may be a TVariable class. TVariable class may be based on a TBasicBuffer template class, which may be part of a libSMARTIO (Sensory-Motor Asynchronous Real Time Input Output) library that provides the basic structure and connectivity between populations in the dynamical system. Other portions of TBasicBuffer may allow the system to acquire sensory input from the environment (TInputBuffer) and deliver action output to the agent (TOutputBuffer), both in real time. In some implementations, TBasicBuffer objects may have a name, a number of elements, an index of a non-CPU device, and may facilitate saving variable values, accessing values for various computations, and/or may have other attributes or methods. In some implementations, a TVariable data structure may include connections to an input source, a number of equations associated with the data structure, an event connection that may look for events on a local receiver and process them, and/or may include other attributes and/or methods.

[0034] TVariable may also provide memory allocation and methods to maintain a population (array) of variables governed by a system of equations. For example, if a dynamical system has a system of equations with variables x1, x2, and x3, an example implementation may use a TVariable-based class. If there are 100 processes in the dynamical system, each of which is governed by the same system of equations with these three variables, then TVariable-based class may contain 100 elements with 3-component system each. Classes derived from TVariable class may perform initialization procedures, assignment of pointers to equations and numerical methods, and implement different computational backends based on available hardware such as GPU or CPU. They may also provide different implementations of the above methods depending on whether the system is used in a standalone or networked mode.

NVIDIA Ex1002 Page 277 of 1081

[0035] Output variables are usually accessed by any element in the system during every time step. The value of the output variable that is accessed by other elements of the system corresponds to the value computed on the previous, not the current, time step. This is realized by dedicating two CPU or GPU memory chunks to output variables—one holds the value computed during the previous time step and is accessible to all computational elements during the current time step, another is not accessible to other elements and is used to accumulate new values for the variable computed during the current time step. In-between time steps these two CPU or GPU memory chunks are switched, so that newly accumulated values serve as accessible input during the next time step, while the old input is replaced with new values of the variable. This switch is implemented by swapping the address pointers to respective CPU or GPU memory chunks. The portion of the time step during which the system computes new values is termed a Compute Semicycle in Figure 2 and the portion of the time step when pointers are switched and new values become available to the rest of the system is termed a Post Semicycle (e.g., see Figure 2).

[0036] Internal variables are computed and used within the same computational element. There is no chance of a race condition in which the value is used before it is computed or after it has already changed on the next time step because within an element the processing is sequential. Therefore, it is possible to compute the new value of internal variable and store it in the same memory location in the memory bank where the old value was read from during the computation.

[0037] During the computation, the master node 102 utilizes network connections with the slave nodes 106 to synchronize the processing across the nodes 102, 106. This corresponds to Compute and Post Semicycles of the example flow chart in Figure 2 and example data flow chart in Figure 3B. In some implementations, the system includes a user interface on the master node that allows direct control of the software operations through computer peripherals such as keyboards, mice, trackballs and other devices. The system can also include an external application 104 that provides information about neural populations. The system may be accessed through API calls by a third party software.

[0038] Slave nodes may have a network manager that establishes point-to-point links between their respective hardware devices for sending and receiving of events. The system may also include a thread manager that controls creation, execution, synchronization and/or destruction of working threads within each slave node to maximize hardware utilization. A

NVIDIA Ex1002 Page 278 of 1081

list of these threads is shown in the Figures 2, 3A, and 3B. The threading structure in the depicted example implementation is based on US Patent No. 8,648,867, which is hereby incorporated herein by reference in its entirety. An example threading structure can include a disk thread that controls the disk output of a node (utilizing a local hard drive of a workstation or a shared disk space of a cluster), opens and closes the files, and does the output for the state variables that are marked for output by the user. Another example thread is communication thread that controls the exchange of output events between neuronal populations or systems of equations within the node and across the network. Other example threads may include non-central processing unit (CPU) computational threads which may be used for computation of neuronal populations on suitable processing devices (e.g., GPUs). In some embodiments, one or more Non-CPU Computational threads are created for each utilized device on the node. Further examples may include CPU Computational threads, which may be used for computation of neuronal populations or systems of equations on available cores of the central processing unit of the slave node. In some implementations, the number of CPU computational threads may be determined by taking the number of cores on the node, subtracting the number of all other threads including the main thread and use the result as the number of CPU Computational threads. In some implementations, for small populations or systems of equations having only one CPU Computational thread might be more efficient.

[0039] Figure 1B illustrates a standalone version 110 in which the control structures of the master node and the computational structures of the slave node are hosted together in (or merged together into) a single software unit that is executed on a standalone workstation or other computer hardware. In such implementations of the system, communications between the systems of equations may happen on the same node. In this implementation of the system a user or external application may initiate the simulation and thread manager directly, and the network management and across-network synchronization may be removed from the system. Everything under the thread manager may remain identical to a slave node 106 in the networked version 100 shown in Figure 1A. In some implementations, an INDRA system may allow switching between instantiating a networked implementation 100 and a standalone version 110 depending on available resources at the moment of instantiation.

[0040] Figure 2 illustrates the control flow through a standalone version of the system (e.g., as shown in Figure 1A). An external application may initiate the engine by starting a simulation and/or thread manager (202). The thread manager may fork and initialize a

NVIDIA Ex1002 Page 279 of 1081

requested number of computational threads plus one disk I/O thread (204) and one communication thread (206). During thread initialization, the thread manager may pass network addresses, number of CPU cores (208) and GPU devices (210) to the communication and/or computational threads on the node as illustrated in Figure 3A. These threads may become initialized and gather on the initialization barrier 212 waiting for the simulation manager to prepare the model and simulation setup. The physical location of the simulation manager may differ between the networked implementation and the standalone implementation. While in the standalone version it may reside on the same hardware as the thread manager; in the networked version the simulation manager may reside on the master node while thread managers may reside only on the slave nodes.

[0041] Model and simulation setup on the system may be accomplished through interaction between the simulation manager with an external application and/or user (214). When the parameterization of the model is substantially complete (216), the user may notify the simulation and thread manager that the model is ready and they may proceed to the initialization barrier. This may allow threads to enter the next stage (e.g., model initializations 218 and 302 in Figures 2 and 3A, respectively). At this stage a model and simulation description may be parsed by the simulation manager (220) on the master node and passed on to various threads on the slave nodes either through network connection if the system is in networked mode or directly if the system is in single node mode. On the slave node, the information is distributed as depicted in Figure 3A.

[0042] Population names are used in disk I/O thread to create corresponding files for the result output (304). List of populations that have outputs is used in communication thread to establish input and output queues on the slave node to communicate between connected populations (306). Equations and relating to the data are communicated to computational threads to instantiate the corresponding data structures (308). On the slave nodes, components of the model may be assigned to different computational threads depending on the hardware configuration of a particular slave.

[0043] After instantiation of the components, the communication queues between these components may be defined (224) as illustrated in Figure 3A (310). In some implementations, this information is passed to computational threads (e.g., model components) and/or to communication threads (e.g., connectivity pattern between components), where the respective data structures are created and initialized 226. After the

NVIDIA Ex1002 Page 280 of 1081

initialization, the initial values of output variables might be passed to disk I/O thread for output as illustrated in Figure 3A (312). Next, disk I/O thread, communication thread, and computational threads may then enter the main simulation loop of the system, so they signal their readiness to the simulation manager and wait on the Post barrier for the Run signal from the simulation manager.

[0044] The main simulation loop of the system may be controlled by an external application on the master node through a set of API commands that it can pass to the simulation manager. The commands may include:

- Quit, which may cause the simulation manager to exit the main loop and inform the slave thread managers locally (standalone version) or through network communication (networked version). The thread managers may inform other threads that the engine is exiting, wait for them to clean up and exit, and then exit the engine.
- Reset, which causes the reset of the components and connections in the model to their initial states. In some implementations, there are several versions of Reset depending on whether the user wants to reinitialize activations (current values) of components, connectivity strengths, or both.
- Re-create, which may cause the engine to return to the model and simulation setup stage and re-parse all the data structures. This command may also redistribute the components of the model to different nodes and/or computing hardware.
- Run, which enters the computational loop. This command may have a parameter that defines whether a user wants a single step though computation, a defined number of steps, or a continuous calculation.
- Stop, which may interrupt the currently executing computational loop after finishing the current cycle.

[0045] In some embodiments, iterations through a computational loop may include two phases or semi-cycles. During the Compute Semicycle, the model components may read their inputs from read-only output memory locations of other components, compute their own new values (228) and write them into their own write-only memory locations. During a Post Semicycle 230, the values from write-only locations in the component may be moved to its read-only locations to be available for others on the next Compute Semicycle. In some implementations, to reduce memory copying, the pointers to read-only and write-only memory locations are interchanged and the actual data is not copied.

NVIDIA Ex1002 Page 281 of 1081

[0046] In some embodiments the communication between neuronal populations or systems of equations during the computational cycles may proceed as shown in Figure 3B. During the Compute Semicycle output events produced in either CPU 312 or non-CPU 314 computational threads are pushed into queues (316) corresponding to these populations' outgoing event queues 232. At the same time a communication thread might receive events (318) from other nodes as well as from local output event queues and insert them into corresponding input event queues. During the Post Semicycle the communication thread delivers (320) the events from input queues to target populations and sends the output queues (322) to disk I/O thread for the output. At the same time the values of the variables can be communicated from the slaves to the master (324) for visualization.

[0047] Figure 3C provides a further elaboration on an example embodiment of the system in terms of event queues engaged by a single population on a single slave node, while Figure 3D shows the same communication scheme from the network standpoint for a slave node with multiple populations. Each neuronal population or system of equations has one outgoing queue termed a local axonal queue 326, where the events produced by this population 328 are inserted during a first Compute Semicycle 303 after synchronization 301. In some implementations, local (outgoing) axonal queues can be implemented with a standard template library class called multimap that holds key-value pairs. For this queue the key may be event production time and the value can include, for example, receiver address and output value. STL multimap may facilitate automatic sorting by key, so this queue may have all output events sorted by their production time. For each receiver of events produced by this population, there is also a shadow axonal queue 330 located on the node where the receiver resides, which may be implemented in a manner similar to local axonal queues. To save memory, there may be only one shadow axonal queue per sender per node, from which all receivers may receive their respective input events coming from this particular sender. During the first Post Semicycle 303, the events are sent from the local axonal queues 326 to shadow axonal queues 330. During a first Compute Semicycles 305 following the first Post Semicycle 303, the events are collected by each node's shadow axonal queue 332. During the next Post Semicycle 307, the events are delivered from shadow axonal queues to corresponding receivers through input (synaptic) queues 334, which may be implemented in a manner similar to local axonal queues, but which may not require a receiver address, may represent timing via delivery time rather than production time, and may adjust the value according to the parameters of the synapse. As a result of this scheme, each output event can

NVIDIA Ex1002 Page 282 of 1081

be delivered to the shadow axonal queues 330 as early as two cycles from its production (i.e., at the end of a second Compute Semicycle 309). The length of each semicycle in real time is determined by the computational load on the system, network load and latency, and may be bound from below by the slowest component. For simple models without much computational load using a standalone system, the semicycle length can be as short as 10 microseconds per a pair of Compute and Post Semicycles.

[0048] In the case of dynamical systems, however, especially those simulating neural populations, there can be axonal delay, and events are often delivered on the millisecond time scale, while the time steps of numerical integration of these equations are significantly shorter for reasons of numerical stability. Thus event delivery often may happen many more than two cycles from their production. The example system presented herein takes advantage of that by bundling the events between nodes in packages and sending these packages over the network only when the first event in the package is due for the delivery. Bundling and sending packages reduces network communication overhead.

[0049] The CPU based computational threads may be organized differently from non-CPU based threads. A component of the model that represents a massive data-parallel population may be passed as a whole to non-CPU thread and to the hardware device it controls to reduce cross-device communication. In some embodiments, the CPU thread population may be split into as many parts as there are CPU threads to balance the CPU load. Thus, initialization and posting for CPU populations may happen as a whole population across threads operation, while computation is done in parts by respective CPUs.

[0050] In some implementations some slave nodes can reside in a cloud environment and other slave nodes can reside on individual robots, while a master node can reside on a user device, such as a tablet device, smartphone, laptop, or desktop computer. This may enable running large-scale robotic artificial brains, where part of the computation occurs on the processors on board of the robot (e.g., a CPU and a GPU on a Single Board Computer(SBC); another example can be a portable handheld device installed onboard and controlling the robot with their respective CPU and GPU), and part of the computation occurs on a server connected via a tethered or wireless link. In some implementations the on-board computation device may be referred to as a "Local brain" and the remote computation device may be referred to as a "Cloud brain" (e.g., see Figure 4), but in both cases these may be hosting slave nodes, while the master node may allow the user to control the system remotely.

NVIDIA Ex1002 Page 283 of 1081

[0051] Referring to Figure 4, the Local brain 400 includes (but is not limited to) a GPU 410, which includes in turn a GPU processor 420 and associated GPU registers 430 and GPU memory 440, a Main Memory 450, a CPU 470, and a Network adapter460.

[0052] The Cloud brain 480 includes (but is not limited to) one or more interconnected computers whether encased separately or within a cluster rack, each of which may include a GPU 482, which includes in turn a GPU processor 484 and associated GPU registers 486 and GPU memory 488, a Main Memory 490, a CPU 492, and a Network adapter 494.

[0053] In general, the Cloud brain 480 has a larger computational capacity with respect to the Local brain 400. Communication between the Local brain 400 and the Cloud brain 480 is established through the network using existing protocols which include but are not limited to Wi-Fi, Bluetooth, and others, and allows for intermediary routers that will convert wired communication with the Cloud brain 480 into wireless communication with the mobile robot. On top of these communication protocols there may be a communication layer that allows information passing between slave nodes described above.

[0054] Neural models can be distributed between slaves in the Local brain 400 and the Cloud brain 480 based on several considerations, for instance, to process some of the neural stages on board in order to reduce network communication load, or minimize the reaction time of the robot (e.g., computing optic flow for collision avoidance).

[0055] A cloud brain 480 may connect to one or more robots. Figure 5 illustrates the case of one slave node, but the method illusirated below can be generalized to an arbitrary number of slave nodes.

[0056] Figure 5 illustrates an example of how Local brains can update the knowledge of the Cloud brain, and how the resulting collective knowledge can be then used by each individual Local brain. The examples below pertain visual input, but the same principle can be applied to all sensory modalities and in general to any neural network stage present in the Robot brain.

[0057] In Figure 5, Local brains 1 (500a), 2 (500b), and 3 (500c) are running on Robot 1, 2, and 3, respectively, in order to learn the identity of three different objects. Robot 1 (502a), 2 (502b) and 3 (502c) can belong to the same user, or to respective users. Since, in some implementations, the robots may be sharing knowledge, the robots' users may, *a priori*, agree to share it. Such implementations may be useful to users within a company where all

NVIDIA Ex1002 Page 284 of 1081

robots share company-specific knowledge, as a non-limiting example. In this example, the Cloud brain 504, which can reside in one or multiple slave nodes, has a Main Memory 506 connected to a plurality of CPU or non-CPU devices 508a-e, and the resulting brain jointly learns the identity of the objects encountered by each robot. Given each Robot is controlled by the Cloud Brain, the knowledge gained via one Robot input is available to all Robot connected to the same Cloud brain.

[0058] Figure 6 illustrates an example of an aggregate Cloud brain 602, where there exists a physical separation of the processors 610 and the memory 606, which includes memory blocks 608a–608c (collectively, dedicated memory blocks 608) and processors 604a–604f (collectively, processors 604) that form respective cloud brains 610a–610c (collectively, cloud brains 610) dedicated to each Robot 600a–600c. This version can used by users who do not wish to share the robotic knowledge on a constant basis. In some implementations sharing may still be possible through specific requests, e.g., if all of the users involved in the exchange of information agree to share. In this example, each Robot 600 may have a separate set of processors 604a–604f and associated memory blocks 608a–608c available that may not overlap with other Robots.

[0059] Figure 7 illustrates an example of how the knowledge derived from the portion of the Cloud brain 602 dedicated to a robot 600 can be transferred to another robot by copying and pasting 708 the model architecture and the associated weight matrices processed by the corresponding processors 604 and stored in the respective dedicated memory block 608 or the Main memory 606.

[0060] Figure 8 illustrates an example of how dedicated Cloud brains can share knowledge without copying or pasting network structure or weights. In the example, a dedicated Cloud brain 610a has been exposed to the visual presentation of three cars, coded by the network represented in 816, which includes neurons coding for different views of the cars (e.g., Object views 818), of the collective object "Car" (e.g., Object category 820), and the representation of an associated verbal label to this object (e.g., "Car" 822). These neural representations stored in Main memory 606 of the dedicated Cloud brain 610a and processed by processors 604a and 604b can be reactivated, for instance in a "sleep" cycle (or off-cycle, namely when the Robot is not active), in the following way:

- A "Car" label is activated in the Cloud brain 610a;

NVIDIA Ex1002 Page 285 of 1081

- This leads to the activation of 1 or more Object categories in Cloud brain 610a associated with that name;
- Each Object category has 1 or more Object views, represented the neural representation of an object associated with that object view, e.g. multiple views of a car (in the figure, only one view of each car class is represented);
- Each Object view and its associated label are then presented to the Cloud brain 610c;
- In the the Cloud brain 610c, previous knowledge was already present in 814 (in the example, the "Car" category already included one example);
- As a result of this learning, the Cloud brain 610c now has knowledge of all objects learned from Cloud brain 610a plus its own knowledge (814), and can further process the information via processors 610e and 610f.

[0061] The same process can be applied for different data samples of the same object or environment, e.g., when different Robots have collected several views of the same object from different angles, or when several robots have explored the same environment with different paths.

[0062] Conclusion

[0063] While various inventive embodiments have been described and illustrated herein, those of ordinary skill in the art will readily envision a variety of other means and/or structures for performing the function and/or obtaining the results and/or one or more of the advantages described herein, and each of such variations and/or modifications is deemed to be within the scope of the inventive embodiments described herein. More generally, those skilled in the art will readily appreciate that all parameters, dimensions, materials, and configurations described herein are meant to be exemplary and that the actual parameters, dimensions, materials, and/or configurations will depend upon the specific application or applications for which the inventive teachings is/are used. Those skilled in the art will recognize, or be able to ascertain using no more than routine experimentation, many equivalents to the specific inventive embodiments described herein. It is, therefore, to be understood that the foregoing embodiments are presented by way of example only and that, within the scope of the appended claims and equivalents thereto, inventive embodiments may be practiced otherwise than as specifically described and claimed. Inventive embodiments of the present disclosure are directed to each individual feature, system, article, material, kit, and/or method described herein. In addition, any combination of two or more such features, systems, articles, materials, kits, and/or methods, if such features, systems, articles, materials,

NVIDIA Ex1002 Page 286 of 1081

kits, and/or methods are not mutually inconsistent, is included within the inventive scope of the present disclosure.

[0064] The above-described embodiments can be implemented in any of numerous ways. For example, embodiments of designing and making the coupling structures and diffractive optical elements disclosed herein may be implemented using hardware, software or a combination thereof. When implemented in software, the software code can be executed on any suitable processor or collection of processors, whether provided in a single computer or distributed among multiple computers.

[0065] Further, it should be appreciated that a computer may be embodied in any of a number of forms, such as a rack-mounted computer, a desktop computer, a laptop computer, or a tablet computer. Additionally, a computer may be embedded in a device not generally regarded as a computer but with suitable processing capabilities, including a Personal Digital Assistant (PDA), a smart phone or any other suitable portable or fixed electronic device.

[0066] Also, a computer may have one or more input and output devices. These devices can be used, among other things, to present a user interface. Examples of output devices that can be used to provide a user interface include printers or display screens for visual presentation of output and speakers or other sound generating devices for audible presentation of output. Examples of input devices that can be used for a user interface include keyboards, and pointing devices, such as mice, touch pads, and digitizing tablets. As another example, a computer may receive input information through speech recognition or in other audible format.

[0067] Such computers may be interconnected by one or more networks in any suitable form, including a local area network or a wide area network, such as an enterprise network, and intelligent network (IN) or the Internet. Such networks may be based on any suitable technology and may operate according to any suitable protocol and may include wireless networks, wired networks or fiber optic networks.

[0068] The various methods or processes (e.g., of designing and making the coupling structures and diffractive optical elements disclosed above) outlined herein may be coded as software that is executable on one or more processors that employ any one of a variety of operating systems or platforms. Additionally, such software may be written using any of a number of suitable programming languages and/or programming or scripting tools, and also

NVIDIA Ex1002 Page 287 of 1081

may be compiled as executable machine language code or intermediate code that is executed on a framework or virtual machine.

[0069] In this respect, various inventive concepts may be embodied as a computer readable storage medium (or multiple computer readable storage media) (e.g., a computer memory, one or more floppy discs, compact discs, optical discs, magnetic tapes, flash memories, circuit configurations in Field Programmable Gate Arrays or other semiconductor devices, or other non-transitory medium or tangible computer storage medium) encoded with one or more programs that, when executed on one or more computers or other processors, perform methods that implement the various embodiments of the invention discussed above. The computer readable medium or media can be transportable, such that the program or programs stored thereon can be loaded onto one or more different computers or other processors to implement various aspects of the present invention as discussed above.

[0070] The terms "program" or "software" are used herein in a generic sense to refer to any type of computer code or set of computer-executable instructions that can be employed to program a computer or other processor to implement various aspects of embodiments as discussed above. Additionally, it should be appreciated that according to one aspect, one or more computer programs that when executed perform methods of the present invention need not reside on a single computer or processor, but may be distributed in a modular fashion amongst a number of different computers or processors to implement various aspects of the present invention.

[0071] Computer-executable instructions may be in many forms, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically the functionality of the program modules may be combined or distributed as desired in various embodiments.

[0072] Also, data structures may be stored in computer-readable media in any suitable form. For simplicity of illustration, data structures may be shown to have fields that are related through location in the data structure. Such relationships may likewise be achieved by assigning storage for the fields with locations in a computer-readable medium that convey relationship between the fields. However, any suitable mechanism may be used to establish a relationship between information in fields of a data structure, including through the use of pointers, tags or other mechanisms that establish relationship between data elements.

NVIDIA Ex1002 Page 288 of 1081

[0073] Also, various inventive concepts may be embodied as one or more methods, of which an example has been provided. The acts performed as part of the method may be ordered in any suitable way. Accordingly, embodiments may be constructed in which acts are performed in an order different than illustrated, which may include performing some acts simultaneously, even though shown as sequential acts in illustrative embodiments.

[0074] All definitions, as defined and used herein, should be understood to control over dictionary definitions, definitions in documents incorporated by reference, and/or ordinary meanings of the defined terms.

[0075] The indefinite articles "a" and "an," as used herein in the specification and in the claims, unless clearly indicated to the contrary, should be understood to mean "at least one."

[0076] The phrase "and/or," as used herein in the specification and in the claims, should be understood to mean "either or both" of the elements so conjoined, i.e., elements that are conjunctively present in some cases and disjunctively present in other cases. Multiple elements listed with "and/or" should be construed in the same fashion, i.e., "one or more" of the elements so conjoined. Other elements may optionally be present other than the elements specifically identified by the "and/or" clause, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, a reference to "A and/or B", when used in conjunction with open-ended language such as "comprising" can refer, in one embodiment, to A only (optionally including elements other than B); in another embodiment, to B only (optionally including elements other than A); in yet another embodiment, to both A and B (optionally including other elements); etc.

[0077] As used herein in the specification and in the claims, "or" should be understood to have the same meaning as "and/or" as defined above. For example, when separating items in a list, "or" or "and/or" shall be interpreted as being inclusive, i.e., the inclusion of at least one, but also including more than one, of a number or list of elements, and, optionally, additional unlisted items. Only terms clearly indicated to the contrary, such as "only one of" or "exactly one of," or, when used in the claims, "consisting of," will refer to the inclusion of exactly one element of a number or list of elements. In general, the term "or" as used herein shall only be interpreted as indicating exclusive alternatives (i.e. "one or the other but not both") when preceded by terms of exclusivity, such as "either," "one of," "only one of," or "exactly one of." "Consisting essentially of," when used in the claims, shall have its ordinary meaning as used in the field of patent law.

NVIDIA Ex1002 Page 289 of 1081

[0078] As used herein in the specification and in the claims, the phrase "at least one," in reference to a list of one or more elements, should be understood to mean at least one element selected from any one or more of the elements in the list of elements, but not necessarily including at least one of each and every element specifically listed within the list of elements and not excluding any combinations of elements in the list of elements. This definition also allows that elements may optionally be present other than the elements specifically identified within the list of elements to which the phrase "at least one" refers, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, "at least one of A and B" (or, equivalently, "at least one of A or B," or, equivalently "at least one of A and/or B") can refer, in one embodiment, to at least one, optionally including more than one, A, with no B present (and optionally including elements other than B); in another embodiment, to at least one, optionally including more than one, B, with no A present (and optionally including elements other than A); in yet another embodiment, to at least one, optionally including more than one, A, and at least one, optionally including more than one, B (and optionally including other elements); etc.

[0079] In the claims, as well as in the specification above, all transitional phrases such as "comprising," "including," "carrying," "having," "containing," "involving," "holding," "composed of," and the like are to be understood to be open-ended, i.e., to mean including but not limited to. Only the transitional phrases "consisting of" and "consisting essentially of" shall be closed or semi-closed transitional phrases, respectively, as set forth in the United States Patent Office Manual of Patent Examining Procedures, Section 2111.03.

NVIDIA Ex1002 Page 290 of 1081

CLAIMS

1. A method of processing sensory data in parallel, comprising:

(A) receiving, at a master node, a request to process incoming sensor data;

(B) receiving, from the master node, a selection of a set of slave nodes, the set of slave nodes selected from a set of nodes capable of processing the incoming sensor data;

(C) establishing a connection between the master node and the set of slave nodes; and, for at least one slave node in the set of slave nodes:

(D) determining a dynamic system model for a portion of the incoming sensor data received by the at least one slave node;

(E) distributing, from the master node to the at least one slave node, a representation of the dynamic system model and a representation of corresponding graph connections between slave nodes in the set of slave nodes;

(F) instantiating at least one local data structure for processing the incoming sensor data based at least in part upon the representation of the dynamic system model distributed in (E);

(G) connecting the at least one slave node to other slave nodes in the set of slave nodes according to the representation of the graph connections distributed in (E);

(H) receiving a signal from the master node to process the incoming sensor data; and

(I) iteratively processing the incoming sensor data using the at least one local data structure instantiated in (F).

2. The method of claim 1, wherein (B) further comprises selecting the set of slave nodes automatically.

3. The method of claim 1, wherein (B) further comprises selecting the set of slave nodes according to user input.

4. The method of claim 1, wherein (E) further comprises:

assigning at least one system of equations corresponding to the dynamical system model to the at least one slave node based on at least one of: computational power of the at least one slave node, computational complexity of the at least one system of equations; and

NVIDIA Ex1002 Page 291 of 1081

communication between parts of a dynamical system represented by the dynamical system model.

5. The method of claim 4, wherein (E) further comprises:

assigning, by the at least one slave node, a set of computational threads for processing at least a portion of the incoming sensor data.

6. The method of claim 1, wherein (G) further comprises:

(G1) determining an operating mode of the master node; and

(G2) connecting the master node to the at least one slave node based on the operating mode of the master node.

7. The method of claim 6, wherein (C) comprises:

connecting the master node to the at least one slave node via a network connection if the master node is configured to operate in a networked mode; and

compiling the slave node into the master node if the master node is configured to operate in a standalone mode.

8. The method of claim 6, wherein the master node is configured to operate in a networked mode or a standalone mode based on available resources.

9. The method of claim 8, wherein the master node is configured to operate in the networked mode if (i) the master node is able to connect to a network and (ii) there is at least one slave node connected to the network.

10. The method of claim 1, further comprising:

(J) storing results of the iterative processing in a data queue in the at least one slave node for communication with the other slave nodes in the set of slave nodes.

11. The method of claim 1, further comprising:

(K) storing results of the iterative processing in a storage medium for off-line analysis.

12. A system for processing and forwarding slave node data, the system comprising:

(A) a first slave node operatively coupled to:

(A1) an outgoing axonal queue associated with a system of equations processed on the first slave node;

NVIDIA Ex1002 Page 292 of 1081

(A2) a shadow axonal queue corresponding to respective outgoing axonal queues of other slave nodes configured to provide input data to the first slave node;

(A3) a set of synaptic queues corresponding to the system of equations processed on the first slave node, the set of synaptic queues configured to preprocess the input data received from the shadow axonal queue; and

(A4) a processing device, operably coupled to the set of synaptic queues, to apply the system of equations to the input data preprocessed by the synaptic queue so as to generate at least one output data value; and

(B) a local outgoing axonal queue, operatively coupled to the processing device, to receive the at least one output data value from the processing device and to provide the at least one output data value to a second shadow axonal queue in at least one of the first slave node or a second slave node.

13. The system of claim 12, wherein the shadow axonal queue is configured to receive at least one data value from a corresponding outgoing axonal queue operatively coupled to a slave node.

14. A system for aggregating and processing sensor data from a robot comprising a local processor that implements a local brain, the system comprising:

at least one slave node, located on a computer network, to implement a cloud brain for receiving and processing the sensor data from the robot, the first processor including:

a main memory module to store the sensor data from the robot, and

a communications device, operably coupled to the main memory module, to transmit at least a portion of the sensor data to the local brain implemented in the local processor of the robot and to receive, from the local brain, at least one data value based at least in part on the at least a portion of the sensor data and on a current state of the local brain.

15. The system of claim 14, wherein the main memory module stores a representation of at least one of:

an outgoing axonal queue associated with a system of equations processed on a first slave node;

a shadow axonal queue corresponding to respective outgoing axonal queue of a second slave node configured to provide input data to the first slave node;

NVIDIA Ex1002 Page 293 of 1081

a synaptic queues corresponding to the system of equations processed on the first slave node, the set of synaptic queues configured to preprocess the input data received from the shadow axonal queue; and

a data structure for processing the sensor data from the robot.

16. The system of claim 14, wherein the communications device is operably coupled to another local brain implemented in another robot and configured to transmit the at least one data value to the other local brain.

17. The system of claim 16, wherein the communications device is further configured to transmit data from a dedicated memory block in the main memory module to the other local brain and to receive, from the other local brain, at least one other data value based at least in part on the data from the dedicated memory block.

NVIDIA Ex1002 Page 294 of 1081

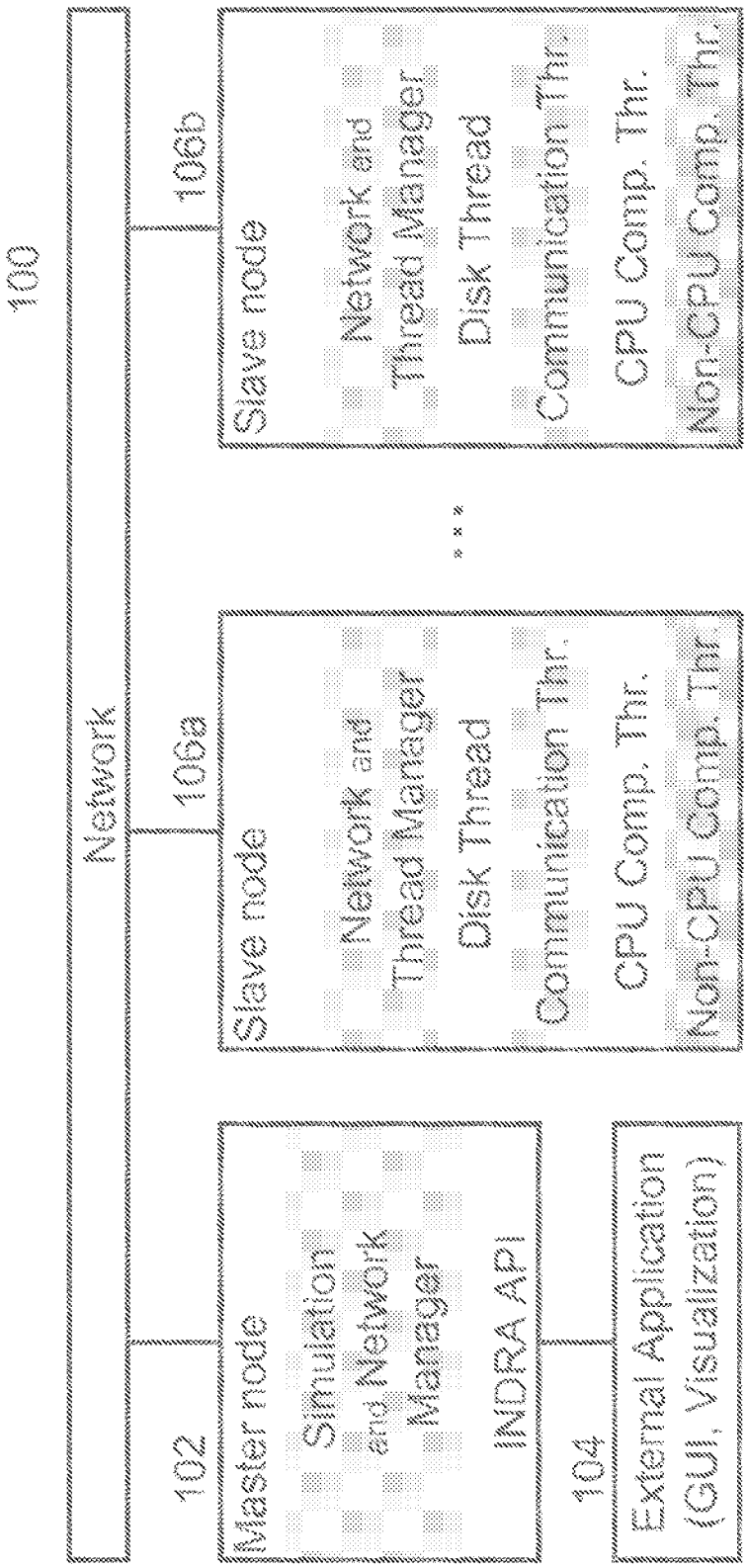

100
Network
102
Master node
Simulation and Network Manager
IMDRA API
104
External Application (GUI, Visualization)
106a
Slave node
Network and Thread Manager
Disk Thread
Communication Thr.
CPU Comp. Thr.
Non-CPU Comp. Thr.
106b
Slave node
Network and Thread Manager
Disk Thread
Communication Thr.
CPU Comp. Thr.
Non-CPU Comp. Thr.

Figure 1A

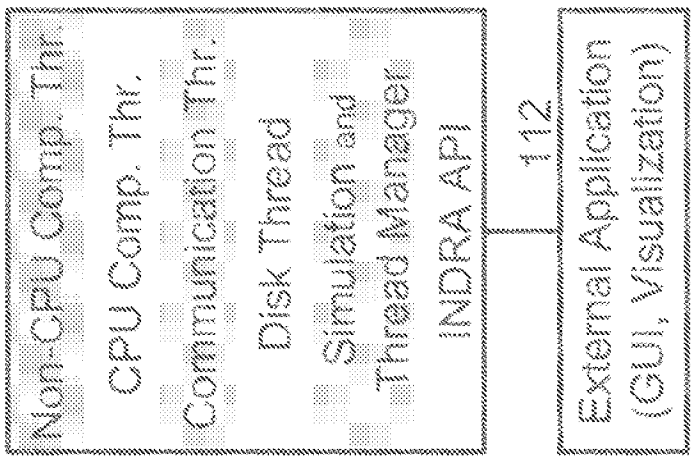

110
Non-CPU Comp. Thr.
CPU Comp. Thr.
Communication Thr.
Disk Thread
Simulation and
Thread Manager
INDRA API
112
External Application
(GUI, Visualization)

Figure 1B

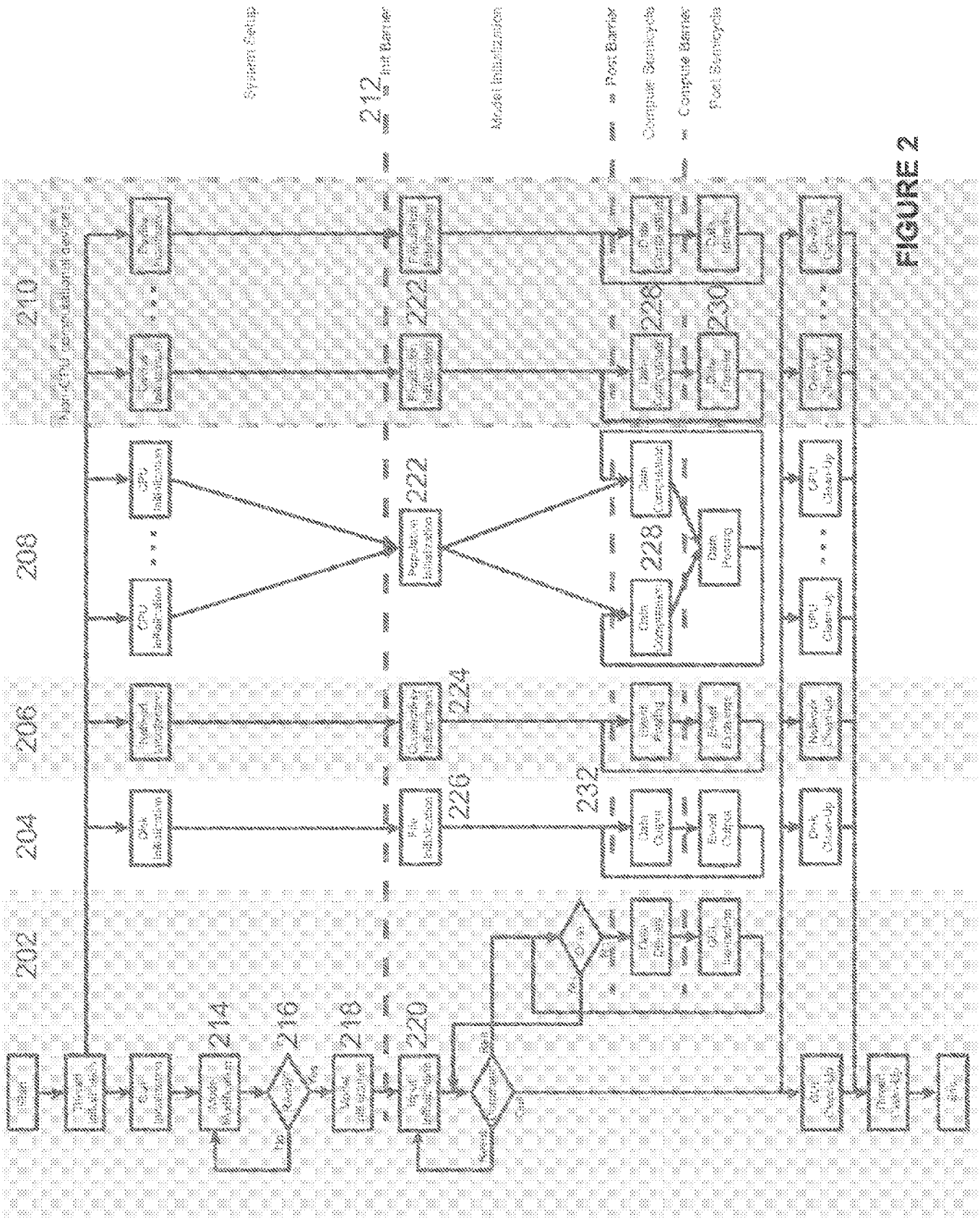

202
204
206
208
210
212
214
216
218
220
222
224
226
228
230
232
System Setup
Model Initialization
= Post Barrier
= Compute Barrier
= Fetch Barrier

FIGURE 2

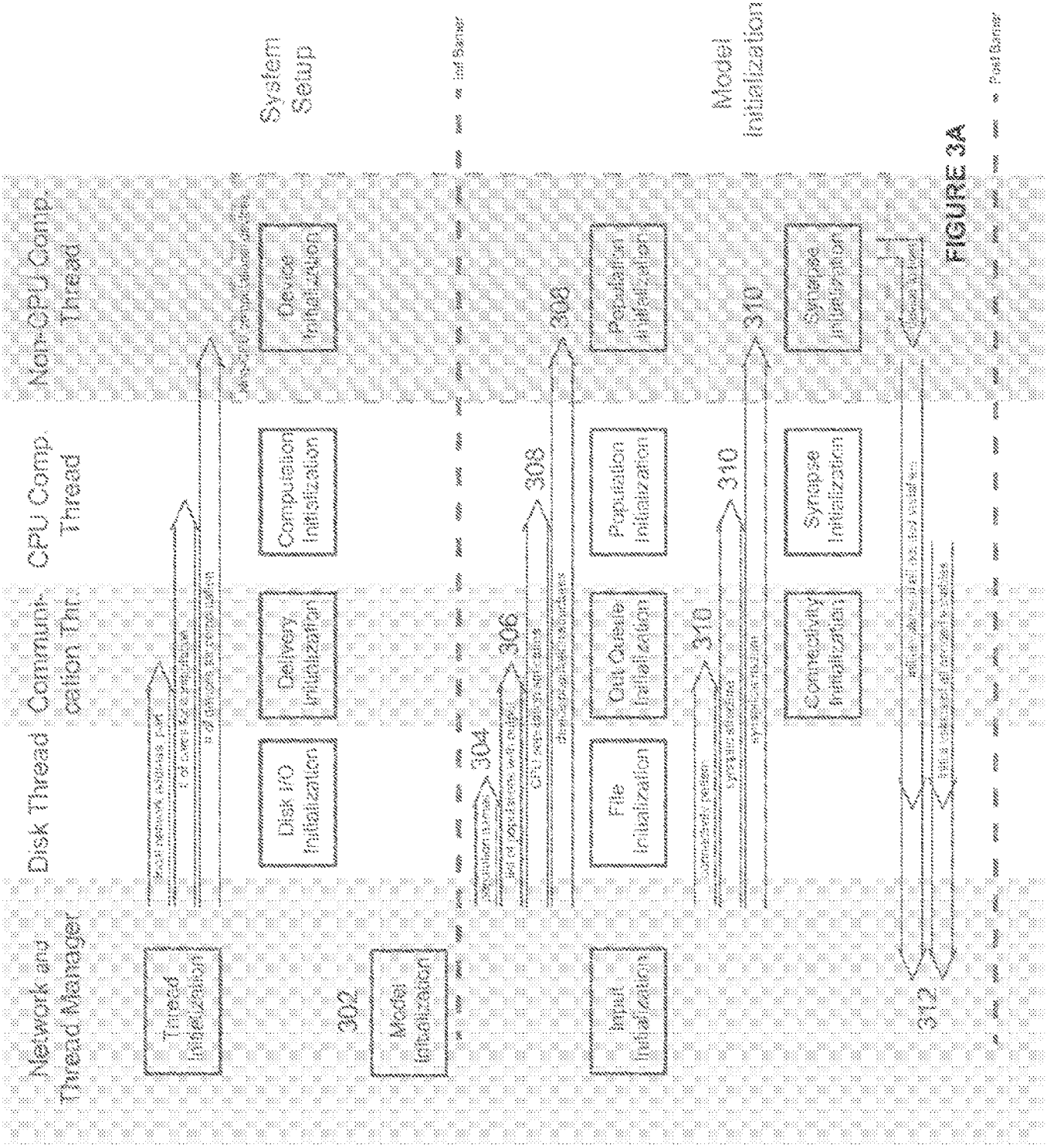

Network and Thread Manager
Disk Thread
Communication Thr.
CPU Comp. Thread
Non-CPU Comp. Thread
System Setup
Model Initialization
Thread Initialization
Disk I/O Initialization
Delivery Initialization
Computation Initialization
Device Initialization
302
Model Initialization
Input Initialization
File Initialization
Out Queue Initialization
Population Initialization
Population Initialization
304
306
308
310
310
310
312
Connectivity Initialization
Synapse Initialization
Synapse Initialization
End Setup
End Sim
Post Sim

FIGURE 3A

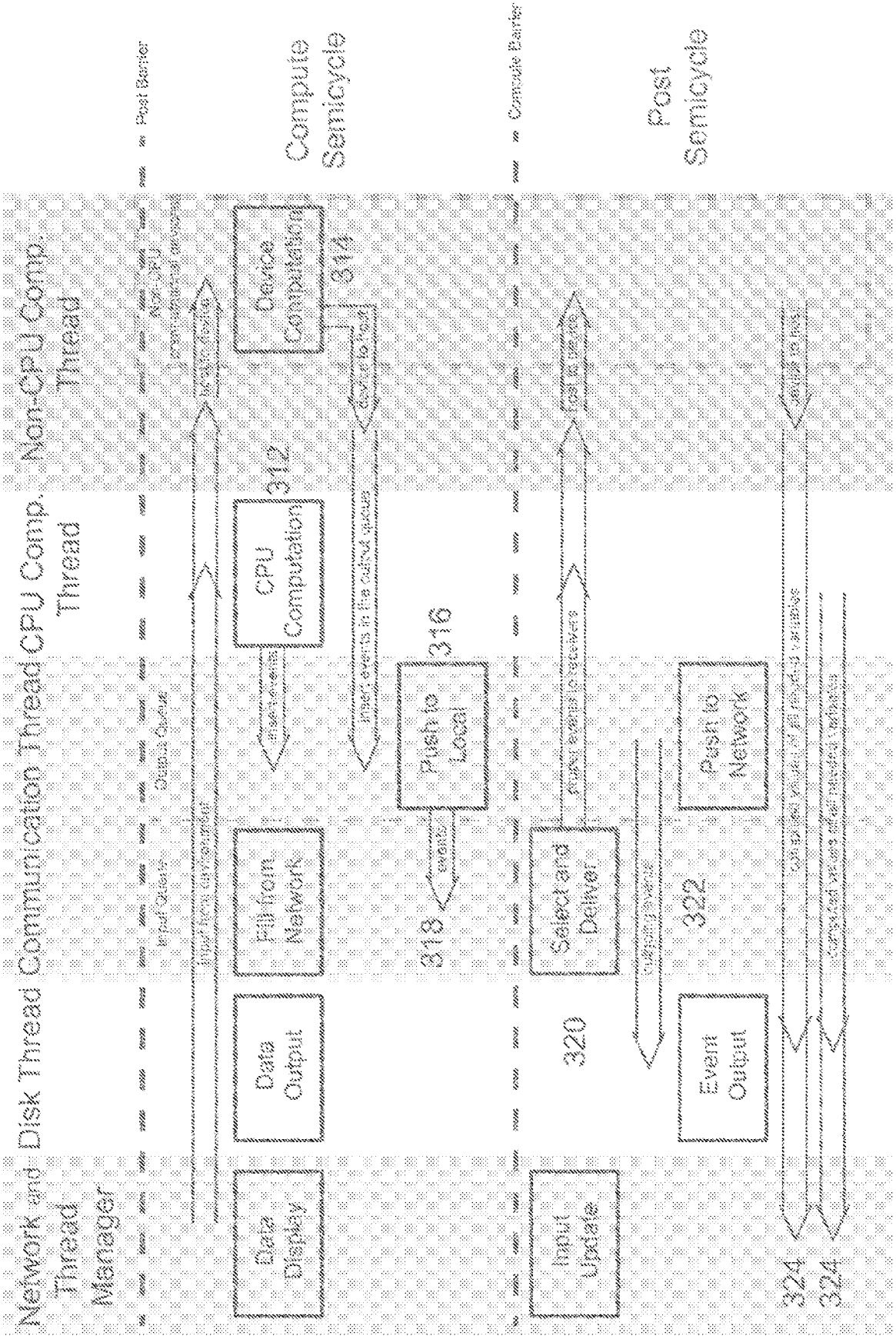

Network and Thread Manager
Disk Thread
Communication Thread
CPU Comp. Thread
Non-CPU Comp. Thread
Post Barrier
Compute Semicycle
Console Barrier
Post Semicycle
Data Display
Input Update
Data Output
Fill from Network
CPU Computation
312
Device Computation
314
Push to Local
316
318
320
Select and Deliver
322
Push to Network
Event Output
324
324

FIGURE 3B

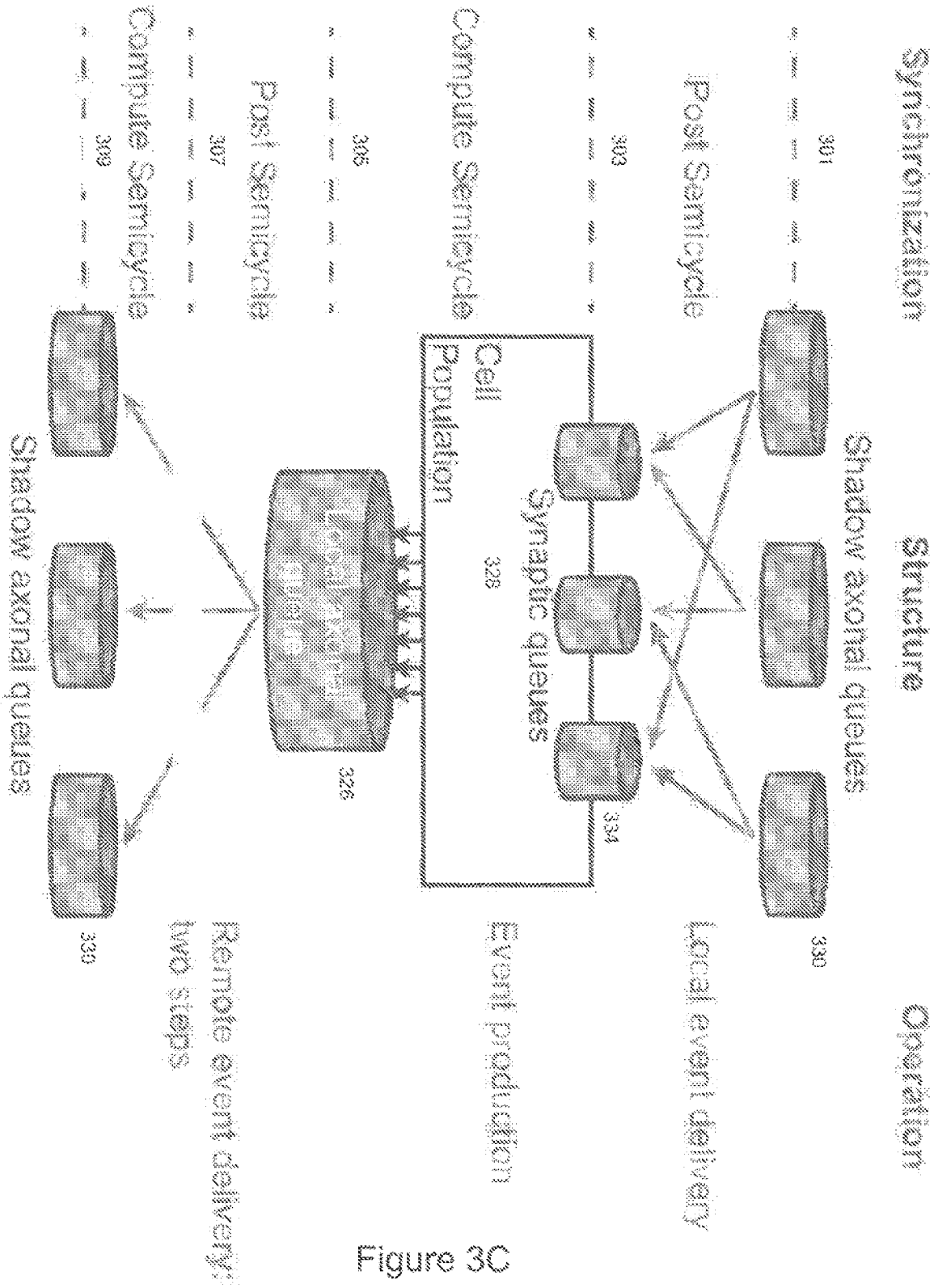

Synchronization
Structure
Operation
Shadow axonal queues
Shadow axonal queues
Post Semicycle
Compute Semicycle
Post Semicycle
Compute Semicycle
Cell population
Synapse queues
Event production
Local event delivery
Remote event delivery
two steps
301
303
305
307
309
326
328
330
332
334
339

Figure 3C

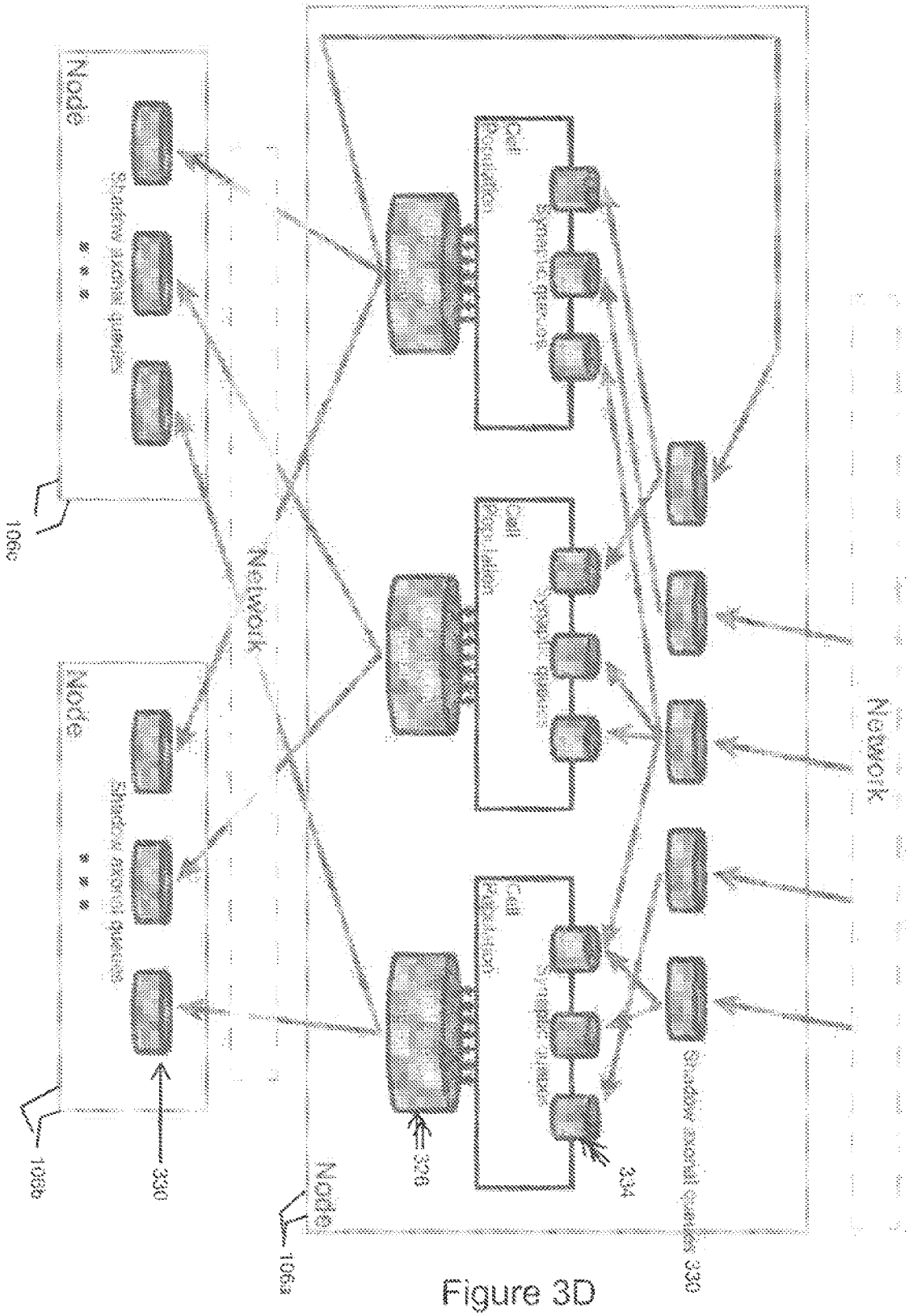


Figure 3D

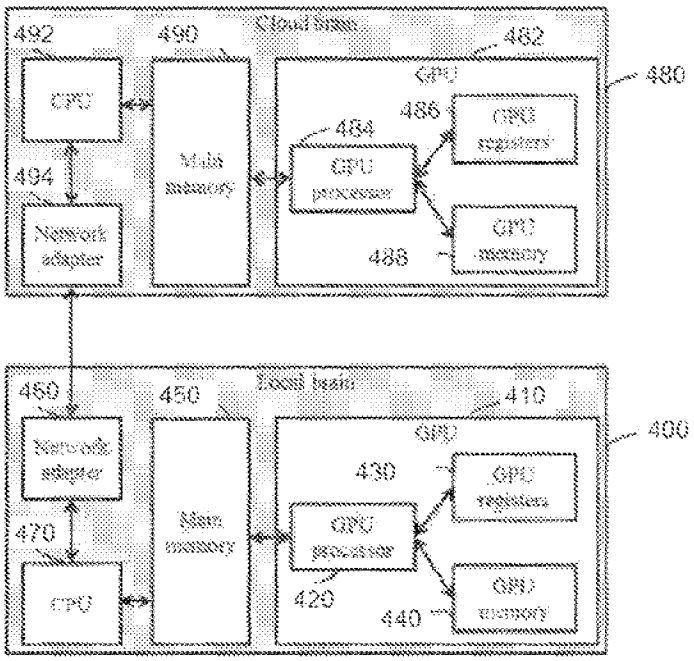

492
490
Cloud brain
482
CPU
480
486
484
494
Main memory
GPU processor
GPU registers
Network adapter
GPU memory
488
460
450
Local brain
410
GPU
400
Network adapter
430
GPU registers
470
Main memory
GPU processor
CPU
420
GPU memory
440

Figure 4

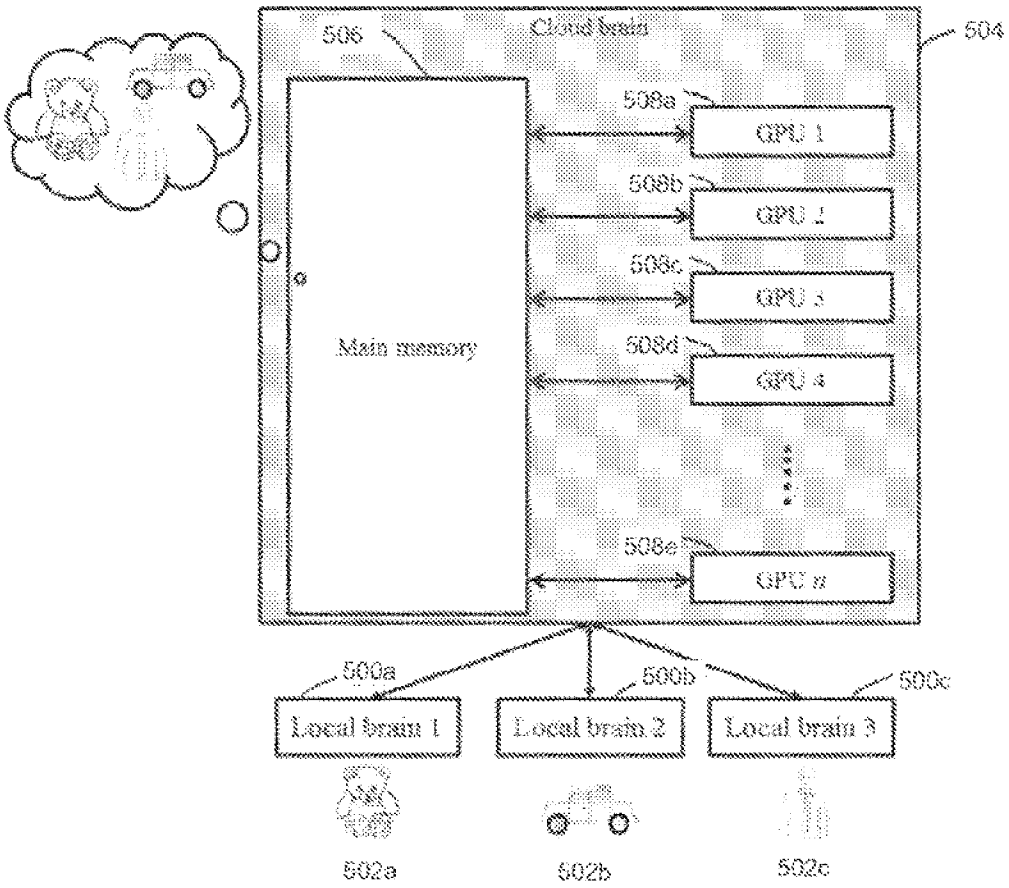

506
Cloud brain
504
508a
GPU 1
508b
GPU 2
508c
GPU 3
Main memory
508d
GPU 4
508e
GPU n
500a
500b
500c
Local brain 1
Local brain 2
Local brain 3
502a
502b
502c

Figure 5

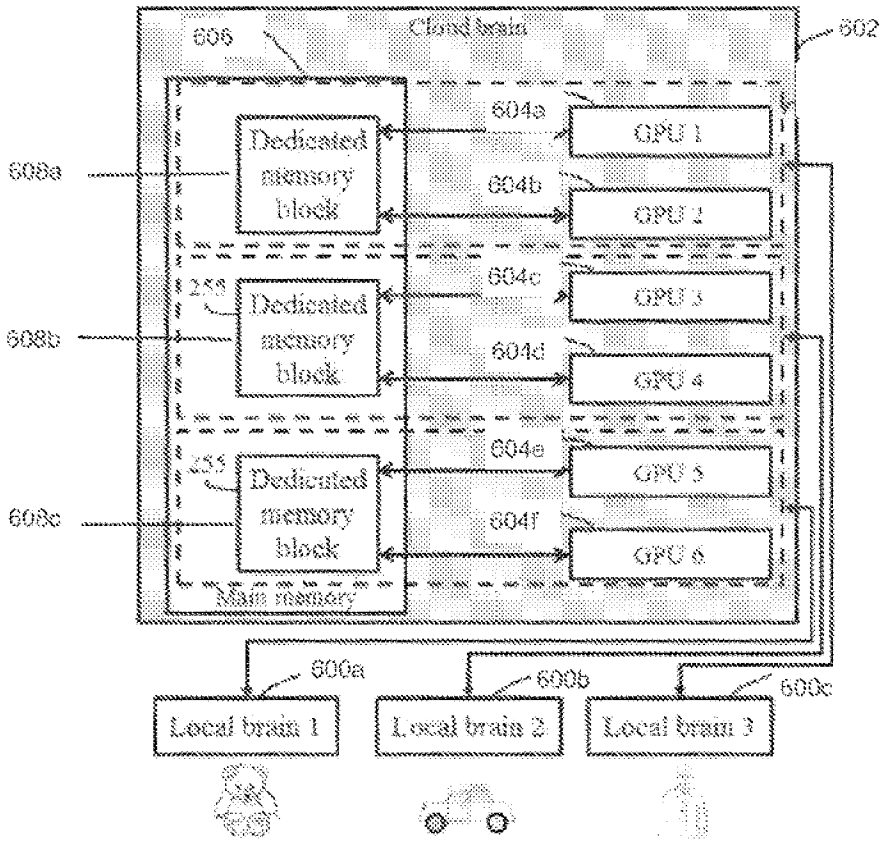

606
Cloud brain
602
604a
GPU 1
608a
Dedicated memory block
604b
GPU 2
604c
GPU 3
608b
Dedicated memory block
604d
GPU 4
604e
GPU 5
608c
Dedicated memory block
604f
GPU 6
Main memory
600a
600b
600c
Local brain 1
Local brain 2
Local brain 3

Figure 6

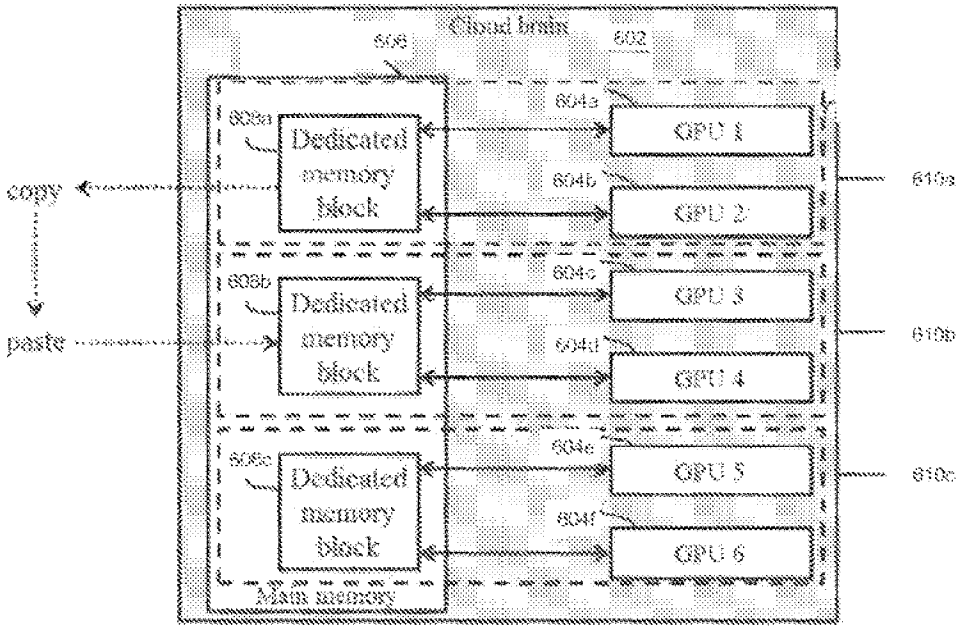

Cloud brain
606
602
608a
Dedicated memory block
608b
Dedicated memory block
608c
Dedicated memory block
Main memory
604a
GPU 1
604b
GPU 2
604c
GPU 3
604d
GPU 4
604e
GPU 5
604f
GPU 6
610a
610b
610c
copy
paste

Figure 7

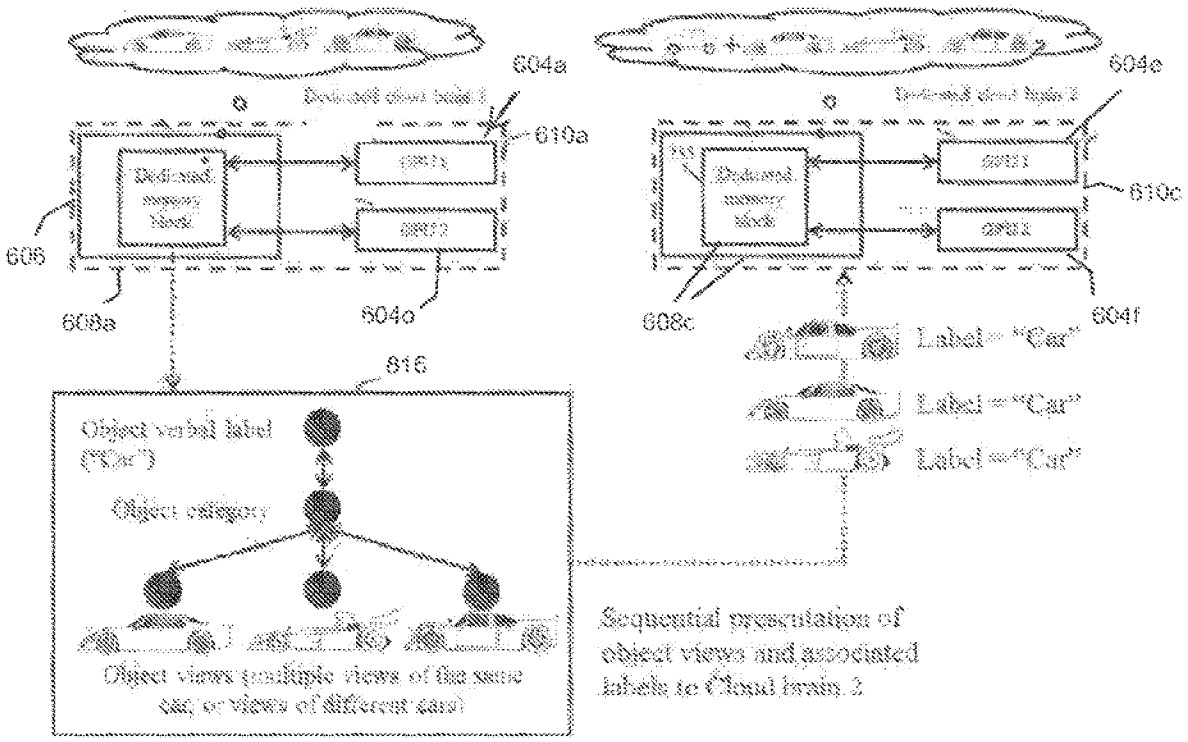

604a
610a
606
608a
604o
604e
610c
608c
604f
816
Label = "Car"
Label = "Car"
Label = "Car"
Object verbal label ("Car")
Object category
Object views (multiple views of the same car, or views of different cars)
Sequential presentation of object views and associated labels to Cloud brain 2

Figure 8

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 September 2015 (24.09.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/143173 A2

(51) International Patent Classification:
G06T 7/20 (2006.01)

(21) International Application Number:
PCT/US2015/021492

(22) International Filing Date:
19 March 2015 (19.03.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/955,755 19 March 2014 (19.03.2014) US

(71) Applicant: **NEURALA, INC.** [US/US]; 8 St. Mary's Street, Suite 613, Boston, Massachusetts 02215 (US).

(72) Inventors: **VERSACE, Massimiliano**; 846 East 3rd Street, Boston, Massachusetts 02127 (US). **GORSHECHNIKOV, Anatoly**; 15 Jefferson Street, Newton, Massachusetts 02458 (US).

(74) Agents: **COLICE, Christopher Max** et al.; 1299 Pennsylvania Avenue, NW, Suite 700, Washington, District of Columbia 20004-2400 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

(54) Title: METHODS AND APPARATUS FOR AUTONOMOUS ROBOTIC CONTROL

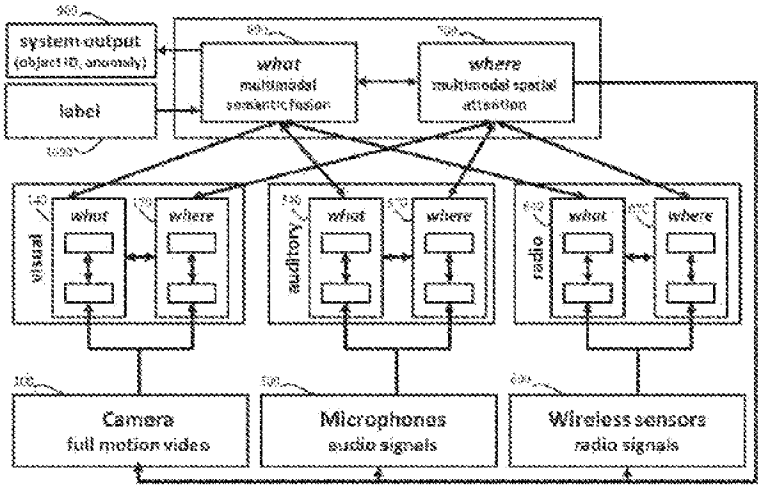


FIG. 1

(57) Abstract: Sensory processing of visual, auditory, and other sensor information (e.g., visual imagery, LIDAR, RADAR) is conventionally based on "stovepiped," or isolated processing, with little interactions between modules. Biological systems, on the other hand, fuse multi-sensory information to identify nearby objects of interest more quickly, more efficiently, and with higher signal-to-noise ratios. Similarly, examples of the OpenSense technology disclosed herein use neurally inspired processing to identify and locate objects in a robot's environment. This enables the robot to navigate its environment more quickly and with lower computational and power requirements.

WO 2015/143173 A2

Methods and Apparatus for Autonomous Robotic Control

CROSS-REFERENCE TO RELATED PATENT APPLICATION

[0001] This application claims priority, under 35 U.S.C. § 119(e), from U.S. Application No. 61/955,755, filed March 19, 2014, and entitled "Methods and Apparatus for Autonomous Robotic Control," which application is hereby incorporated herein by reference in its entirety.

GOVERNMENT SUPPORT

[0002] This invention was made with government support under Contract No. FA8750-12-C-0123 awarded by Air Force Research Laboratory (AFRL) and under Contract No. NNX12CG32P awarded by NASA Phase I STTR. The government has certain rights in the invention.

BACKGROUND

[0003] For a mobile robot to operate autonomously, it should be able to learn about, locate, and possibly avoid objects as it moves within its environment. For example, a ground mobile/air/underwater robot may acquire images of its environment, process them to identify and locate objects, then plot a path around the objects identified in the images. Additionally, such learned objects may be located in a map (e.g., a world-centric, or allocentric human-readable map) for further retrieval in the future, or to provide additional information of what is preset in the environment to the user. In some cases, a mobile robot may include multiple cameras, e.g., to acquire stereoscopic image data that can be used to estimate the range to certain items within its field of view. A mobile robot may also use other sensors, such as RADAR or LIDAR, to acquire additional data about its environment. RADAR is particularly useful for peering through smoke or haze, and lidar returns can sometimes be used determine the composition of objects within the environment.

[0004] A mobile robot may fuse LIDAR, RADAR, IR, ultrasound, and/or other data with visible image data in order to more accurately identify and locate obstacles in its environment. To date, however, sensory processing of visual, auditory, and other sensor information (e.g., LIDAR, RADAR) is conventionally based on "stovepiped," or isolated processing, with little interactions between modules. For this reason, continuous fusion and learning of pertinent information has been an issue. Additionally, learning has been treated

NVIDIA Ex1002 Page 308 of 1081

mostly as an off-line method, which happens in a separate time frame with respect to performance of tasks by the robot.

[0005] As opposed to this, animals perform both learning and performance simultaneously, effortlessly segmenting sensory space is coherent packets to be fused in unique object representations. An example is a conversation between two people in a crowded party, where the signal-to-noise ratio (S/N) of the speaker voice is extremely low. Humans are able to focus visual attention to the speaker, enhance S/N, bind the pitch of the speaker to the appropriate person speaking, and learning the joint "object" (visual appearance and speaker identity) so that recognition of that person is possible with one modeality alone.

SUMMARY

[0006] Embodiments of the present invention include a system for automatically locating and identifying an object in an environment. In one example, the system comprises at least one sensor (e.g., an image sensor, RADAR, microphone, etc.), a spatial attention module (aka a *Where* system) operably coupled to the sensor, and a semantics module (aka a *What* module) operably coupled to the spatial attention module. In operation, the sensor acquires sensor data representing of at least a portion of the object. The spatial attention module produces a foveated representation of the object based on the sensor data, track a position of the object within the environment based on the foveated representation, and selects another portion of the environment to be sensed by the sensor based on the foveated representation of the object. And the semantics module determines an identity of the object based on the foveated representation of the object.

[0007] It should be appreciated that all combinations of the foregoing concepts and additional concepts discussed in greater detail below (provided such concepts are not mutually inconsistent) are contemplated as being part of the inventive subject matter disclosed herein. In particular, all combinations of claimed subject matter appearing at the end of this disclosure are contemplated as being part of the inventive subject matter disclosed herein. It should also be appreciated that terminology explicitly employed herein that also may appear in any disclosure incorporated by reference should be accorded a meaning most consistent with the particular concepts disclosed herein.

NVIDIA Ex1002 Page 309 of 1081

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The skilled artisan will understand that the drawings primarily are for illustrative purposes and are not intended to limit the scope of the inventive subject matter described herein. The drawings are not necessarily to scale; in some instances, various aspects of the inventive subject matter disclosed herein may be shown exaggerated or enlarged in the drawings to facilitate an understanding of different features. In the drawings, like reference characters generally refer to like features (e.g., functionally similar and/or structurally similar elements).

[0009] FIG. 1 shows an example OpenSense architecture, in this case illustrating three sensory modalities, but expandable to other sensor types and number.

[0010] FIG. 2A is a block diagram of an example OpenEye system.

[0011] FIG. 2B is a block diagram of the *Where* Pathway module shown in FIG. 2A.

[0012] FIG. 2C is a block diagram of the *What* Pathway module shown in FIG. 2B.

[0013] FIG. 3 illustrates control of a robot using the OpenEye system via a remote controller, such as a tablet or smartphone.

[0014] FIGS. 4A–4D illustrate a process for identifying and locating objects in a robot's environment by fitting a spatial shroud to successive images of the robot's environment.

[0015] FIGS. 5A and 5B illustrate an implementation of the temporal Adaptive Resonance Theory (tART) model.

[0016] FIGS. 6A and 6B illustrate operation of a high-level *Where* pathway.

[0017] FIG. 7 illustrates anomaly detection based on raw data match/mismatch.

[0018] FIG. 8 illustrates anomaly detection based on raw data match/mismatch.

[0019] FIG. 9 illustrates a search driven by the presence of a search target.

[0020] FIG. 10 illustrates a Virt-U environment that integrates game engine and neural computation environment.

[0021] FIG. 11 illustrates Virt-U operation in simulation mode.

[0022] FIG. 12 illustrates Virt-U operation in "no-brainer" mode.

[0023] FIG. 13 is a diagram of the sensors, actuators, and processors in an example robotic system.

NVIDIA Ex1002 Page 310 of 1081

[0024] FIG. 14 is a diagram of objects within the robotic system and their communication streams.

[0025] FIG. 15 illustrates a fine observer that scans unknown areas and objects in an input image and a coarse observer that bases the fine observer via fast scene segmentation.

[0026] FIG. 16 illustrates coarse and fine observer interactions based on sensor data acquired by a robot.

[0027] FIG. 17 illustrates differences in processing with (center column) and without (left column) a coarse observer and with a coarse observer trained in different environments (right column).

[0028] FIGS. 18A and 18B illustrate an example simulation of autonomous navigation with a coarse observer.

[0029] FIG. 19 illustrates temporal continuity in the fine observer.

DETAILED DESCRIPTION

[0030] The technology described herein provide a unified mechanism for identifying, learning, localizing, and tracking objects in an arbitrary sensory system, including data streams derived from static/pan-tilt cameras (e.g., red-green-blue (RGB) cameras, or other cameras), wireless sensors (e.g., Bluetooth), multi-array microphones, depth sensors, infrared (IR) sensors (e.g., IR laser projectors), monochrome or color CMOS sensors, and mobile robots with similar or other sensors packs (e.g., LIDAR, IR, RADAR), and virtual sensors in virtual environments (e.g., video games or simulated reality), or other networks of sensors. Additionally, the technology disclosed herein allows for stable learning of the identity of single sensor modality of multiple sensor modality objects in the above sensor data streams. Additionally, the technology disclosed herein enables fusion of disparate sensory information in a unified sensory object using spatial information (location of object in 3D space) to (a) enhance sensor information pertinent to the object and suppress sensor information that is not pertinent (S/N enhancement) and (b) learn joint representation of the object via online learning.

[0031] In one example, the the technology disclosed herein processes the input either as one or more continuous streams representing the environment or as static sensors snapshots of the environment. The technology applies a hierarchical neurally-inspired mathematical model that combines several learning systems in a reciprocally connected,

NVIDIA Ex1002 Page 311 of 1081

feedforward/feedback (including recurrent) architecture. This learning technique allows networks of rate-based (neurons or nodes or population of nodes that are represented by continuous variables) or spike-based (neurons or nodes that are represented by continuous variables and that communicate by spikes, or sparse binary events) neural models organized in adaptive (learning) stackable modules to learn, in real time, novel patterns. These techniques do not require batch learning, yet allow fast, incremental, online learning as exhibited in fast learning models.

[0032] The technology presented herein addresses major limitations in current approaches, including but not limited to: (1) the inability to segregate discrete objects of interest to be learned in the data stream from their "background"; (2) the need to design separate sub-systems for object segregation, object recognition, and object tracking; (3) the inability of a system to maintain temporal continuity (identity, position) of objects in the environment taking into account motion of the object and observer; and (4) the need to separate system learning and system use (or deployment, or performance) in two distinct stages to prevent overriding prior learning of object in the data stream.

[0033] Neurally Inspired Robot Perception, Object Identification, and Object Location

[0034] A conventional robot does not perceive its environment like a human. For example, a robot may "see" its environment by acquiring imagery of some or all or its environment at a uniform resolution. It then processes the imagery by dividing the imagery into a grid of pixels and examining each pixel in the grid. This process can take too much time and too much energy to be useful for identifying objects moving relative to the robot, especially if the robot is moving at relatively high velocity (e.g., a drone flying at low altitude). In addition, the robot may spend an inordinate amount of time processing empty or irrelevant pixels.

[0035] A human does not process the detail of entire images on a pixel-by-pixel basis. Instead, the human eye acquires imagery of non-uniform resolution: the central part of the retina, or fovea, which is densely packed with light-sensitive cones, acquires the central part of each image at relatively fine resolution. And the peripheral portion of the retina, which is covered at lower density with light-senstive rods and cones, acquires the peripheral portion of each image at coarser resolution. The resulting "foveated imagery" has resolution that varies spatially across each image, with the finest resolution at a fixation point and coarser

NVIDIA Ex1002 Page 312 of 1081

resolution elsewhere. This notion of obtaining imagery at a resolution that varies spatially across each image is referred to herein as "foveation."

[0036] To account for the spatial variation in image resolution, a human moves his or her eyes rapidly among different points in his or her field of view. For instance, a human may fixate on points at or near an interesting portion of a scene, such as a face, for relatively long periods, and fixate on points at or near less interesting portions of the scene, such as a tree, for shorter periods, if at all. These quick, simultaneous movements to different fixation points, or saccades, allow a human to identify and locate items of interest without spending time or energy examining interesting portions of the scene.

[0037] Similarly, the OpenSense technology disclosed herein allows a robot to identify and locate objects in its environment using "foveated" data collection and "saccade" style allocation of sensor resources as explained below with respect to FIGS. 1–4. For instance, in a visual implementation of OpenSense, called "OpenEye," one or more processors may control collection and processing of visual imagery according to a neural model inspired by the human brain. A camera or other sensor acquires imagery of the robot's environment and passes this imagery to a graphics processing unit (GPU) or other suitable processor, which locates and identifies one or more objects in the imagery (e.g., using the *What* and *Where* pathways described in greater detail below) based on the imagery itself and information about the sensor's orientation, position, and/or field of view. In some cases, the GPU may translate the imagery among different frames of reference, including camera-centered, robot-based egocentric, and allocentric frames of reference, to make processing more efficient and/or more precise.

[0038] The processor also determines the next fixation point of the sensor system based on the location and/or identity of the object(s). In some cases, it transmits movement vector representing the saccade between the current fixation point and the next fixation point to an actuator that then actuates the sensor appropriately. For instance, the processor may cause a pan-tilt actuator to move a camera mounted on the robot so as to acquire imagery of an object from different angles and/or positions. The robot itself may move to change the sensor's field of view. In other cases, the processor may cause synthetic "saccades," e.g., by processing different portions of the same image or different portions of different images at different resolutions depending on the objects and their locations. The robot may also use object information and sensor position and orientation data to inhibit the sensor from fixating repeatedly on the same object or the same portion of the scene.

NVIDIA Ex1002 Page 313 of 1081

[0039] Just like human perception, robotic perception in the OpenSense framework can extend to sources of sensory information besides visible imagery. For instance, the OpenSense framework can be applied to range data acquired by RADAR, LIDAR, and SONAR. It can also be applied to passive electromagnetic sensing, including audio sensing. Moreover, the GPUs and/or other processors can allocate sensor resources dynamically in a manner similar to the foveation and saccading discussed above with respect to visual imagery, e.g., by causing a nodding LIDAR to change its sweep arc or rate, by processing audio data at different spectral resolutions in different bands or detecting sound waves emanating from a particular location, by orienting the receptivity pattern of a radio-frequency in particular direction, etc.

[0040] And like a human brain, the neural-network can fuse data from multiple sources in order to more efficiently identify and locate objects in a robot's environment as explained below with respect to FIG. 1. For example, a robot may use an image sensor to take a picture of an object, then identify and locate the object from the picture using a vision-oriented *What/Where* system. The vision-oriented *What/Where* system sends an output representing the object's identity and/or location (e.g., "focus at x = 12, y = 32, z = 31") to a joint *What/Where* system that also controls an audio-oriented *What/Where* system and a RADAR-oriented *What/Where* system. In some cases, the vision-oriented *What/Where* system's output may be in a frame of reference defined with respect to the robot (an egocentric reference frame) or defined with respect to other objects in the environment (an allocentric reference frame).

[0041] The joint *Where* system tells one or more of the other sensory modules in the OpenSense framework (auditory, RADAR, etc): "all focus at x = 12, y = 32, z = 31." The auditory system responds to this command by suppressing anything in the auditory data stream that is not in x = 12, y = 32, z = 31, e.g., by using Interaural Time Differences (ITD) to pick up signals from one location, and suppress signals from other locations. Similarly, the RADAR system may focus only on data acquired from sources at or near x = 12, y = 32, z = 31, e.g., by processing returns from one or more appropriate azimuths, elevations, and/or range bins.

[0042] Each lower-level *Where* system may generate its own estimate of the object's location and pass this estimate to its corresponding *What* system and to the joint *Where* system. Similary, each lower-level *What* system may generate its own object identification based on the corresponding object location estimate and pass this information to the joint

NVIDIA Ex1002 Page 314 of 1081

What/Where system. The robot's joint *What/Where* fuses and processes this information to identify and locate the object, possibly with a higher degree of confidence than any of the lower level *What/Where* systems. For instance, the joint *Where* system may select a unique spatial location in 3D space, then bias the *What* system module to fuse the identity of separate sensory streams into a coherent object-centered representation.

[0043] Because the technology disclosed herein mimics human neural processing, it can process imagery and other sensory data more efficiently and identify objects in the robot's environment more quickly. This is especially useful for robots in hazardous applications, such as planetary exploration, where processing and battery efficiency are critical, and for robots that collect large volumes of data, such surveillance drones, where efficient sensemaking is key to interpreting large amounts of real-time data. It also has general application to all types of vision systems, including simulations, such as those used in video games, flight simulators, etc.

[0044] **The OpenSense System**

[0045] FIG. 1 illustrates an embodiment of the technology disclosed herein, called OpenSense, which allows real-time sensing and cognitive reasoning on heterogeneous sensor streams. OpenSense can autonomously fuse multiple sensory inputs into a multisensory scene, segregate this multisensory scene into objects that correspond to distinct physical sources, dynamically allocate sensor resources for fast and automatic enhancement of high-priority targets and noise suppression, and detect when anomalous changes occur to known objects based on changes in low-level sensor signature.

[0046] FIG. 1 shows that OpenSense includes several *What-Where* systems (described in greater detail below). FIG. 1 shows how these *What-Where* systems can be combined together in a higher-order sensory processing system (OpenSense) that can fuse data from many sensors into a coherent object, and continuously learn about the coherent object while tracking it. This higher-order sensory processing system goes beyond other systems as it combines online learning, focus of attention (namely, learning only what belongs to objects, and fuse the corresponding data, rather than fuse all which comes to the sensors irrespective of what in the environment generates the signal), and tracking in one single solution.

[0047] Although FIG. 1 illustrates OpenSense system with three sensory inputs, the OpenSense system can be generalized to arbitrary numbers and types of sensory inputs (e.g., static/pan-tilt cameras, wireless sensors, multi-array microphone, depth sensors, IR laser

NVIDIA Ex1002 Page 315 of 1081

projectors, monochrome CMOS sensors, and mobile robots with similar or other sensors packs – e.g., LIDAR, IR, RADAR, and virtual sensors in virtual environments – e.g., video games or simulated reality - or other networks of sensors).

[0048] In the example shown in FIG. 1, camera inputs (100), audio signals (500), and radio signals (600) are collected from a camera, microphones, and radio sensors (e.g., Bluetooth), respectively. Visual, auditory, and radio information is processed by three modules based on the same basic architecture, each including mutual interactions between respective semantic components, also called *What* components (semantic, 140, 540, and 640), and respective spatial attention components, also called *Where* components (spatial attention, 170, 570, and 670). Individual *Where* pathways converge in a high-level *Where* stage (700) and compete to grab "attentional focus" among sensory systems. This high-level *Where* stage (700) allows a high-level *What* system (800) to fuse pertinent multi-sensory information, e.g., creating an objects category that maps spatially-defined visual, auditory, and radio signals in a unique object. The high-level *What* system (800) also projects back to each sensor raw data stages (bi-directional connections) to match object expectation with low-level data and generate anomaly alerts. The high-level *Where* system (700) narrows the sensors' "fields of view" to collect information about the object in the current focus of attention until the high-level *What* system (800) has gained enough data for learning or classification. The system automatically generates scene metadata associated with each video frame summarizing object identity and anomalies (900). Finally, analysts can provide human-readable labels (1000) for the multi-sensory object.

[0049] An implementation of this technology for the visual domain is named OpenEye and uses physical or virtual pan-tilt or static cameras to collect data. OpenEye can be implemented as an artificial, active sensory system that addresses limitations set forth above in a unified framework. OpenEye may be used in both artificial environments (e.g., synthetically generated environments via video-game engine) and natural environments. OpenEye learns incrementally about its visual input, and identifies and categorizes object identities and object positions. OpenEye can operate with or without supervision---it does not require a manual labeling of object(s) of interest to learn object identity. OpenEye can accept user input to verbally label objects. OpenEye simulates a mammalian brain's dorsal (*where* – controlling where to look) and ventral (*what* – controlling the content of the image) pathways by using simulated eye movements (in virtual or real cameras) to learn identity of objects in complex images.

NVIDIA Ex1002 Page 316 of 1081

[0050] In some implementations OpenEye uses a space-variant, log-polar representation of the input visual field to sample the image "view" generated by each eye movement. The log-polar representation provides some invariance to translation/rotation. The log-polar representation can also provide substantial savings in processing time with better scalability to large datasets by employing non-uniform input sampling and rapid scan of image segments, as opposed to processing of the whole image at equal resolution.

[0051] OpenEye uses the *what*-to-*where* feedback to sample the image intelligently. OpenEye does so by using knowledge of the identity of the current object and its context to focus on spatial locations that yield greatest disambiguation of competing object identity (e.g., areas of an image that are more unique to an object). OpenEye may be validated on natural and synthetic images, as well as on the standard datasets (one example is the Mixed National Institute of Standards and Technology (MNIST) handwritten digit dataset).

[0052] As opposed to other approaches (e.g., neural networks), the OpenSense method, and the specific OpenEye implementation , may not need to rely on extensive training (batch training) to be able to classify correctly objects in the data stream. OpenEye can learn new knowledge online (e.g., during performance) without corrupting or forgetting previously learned knowledge and without needing to retrain the system on the whole knowledge database (batch learning). Additionally, the system is able to autonomously search for information in an image via an active visual search process, which mimics the mechanism used by mammals to rapidly and efficiently scan their visual world for information to confirm or disprove the current hypothesis about the object class. The OpenEye memory system allows on-line changes of synaptic weights, which represent the memory (knowledge) of the system. Additionally, OpenEye can mimic human eye movements by reproducing human fixation patterns with or without a training session where OpenEye learns the fixation location of a human user via eye-tracker.

[0053] **Visual Stream Exploration and Visual Object Learning**

[0054] The OpenEye model proposes a method for combining *visual stream exploration* and *visual object learning*. Each is considered separately below.

[0055] **Visual stream exploration models**

[0056] The computational model proposed by Itti and Koch (2001) simulates aspects of human vision which predict the probability that a particular image area will attract an observer's attention and eye movements. The Itti and Koch model includes only bottom-up,

NVIDIA Ex1002 Page 317 of 1081

or sensory features, whereas OpenEye also accounts for cognitive (top-down) biases on eye movements. Additionally, the Itti and Koch model does not include learning, object, or scene recognition, which are instead incorporated in OpenEye, where they bias image stream exploration as discussed below.

[0057] OpenEye also differs from the Riesenhuber and Poggio (1999) neural model, which employs a spatially homogenous representation of the image. In contrast, OpenEye uses both a spatially variant representation of the image and sensor movement. Both the Itti & Koch (2001) and Riesenhuber & Poggio (1999) models postulate that visual objects need to be identified in one glance. OpenEye, instead, accounts for the potential need to explore the input sensory image to gather additional evidence for recognition, which is particularly useful for ambiguous objects/scenes (e.g., occluded objects).

[0058] **Visual object learning models**

[0059] In terms of learning, OpenEye may use two interchangeable learning methodologies. The first method, described in detail below, is based on the Baraldi and Alpaydin (1998, 2002) and Baraldi and Parmiggiani (1997) learning models, which provide the following benefits. The second method is based on a recurrent adaptive architecture described herein. Both methodologies simultaneously implement fast and slow learning.

[0060] Usually, fast learning (e.g., Carpenter and Grossberg, 1987) systems underperform slow-learning ones (Rumelhart *et al.*, 1986), but the former are much more useful in engineered system such as robots or sensors operating in real-time in a rapidly changing environment. After only single instance of presentation of each item, humans and other animals can learn to recognize pictures, words, names, and faces, and recording at a local cellular level confirms that neurons can change to reflect such fast learning (Bunzeck & Düzel, 2006; Rutishauser et al., 2006). To date, no artificial system has been engineered to achieve this goal in a machine.

[0061] Several object recognition algorithms have been developed over the last few decades (for reviews, see Besl and Jain, 1985; Logothetis and Sheinberg, 1996; Riesenhuber and Poggio, 2000; Bengio et al., 2012). In general, a commonality between these algorithms is the focus on finding the appropriate representation for the data, where the difference among algorithms performance is due to the nature of the features/input data transformations. For instance, convolutional network models (Ranzato et al., 2007; Jarrett et al. 2009; LeCun et al., 2010) and restricted Boltzmann machines (Smolensky,1986; Salakhutdinov and

NVIDIA Ex1002 Page 318 of 1081

Hinton, 2009) are among the best object recognition algorithms. Both classes of algorithms perform three main steps: (1) feature extraction, which can be either hardwired, random, or learned; (2) non-linear transformation on the resulting filtered data; and (3) a pooling step on the result of step (2). The connectivity between stages and the number of filter-transform-pool stages can vary.

[0062] Deep learning networks include networks where there are several layers of stacked filter-transform-pool, e.g., in the HMAX model (Riesenhuber & Poggio, 1999) and deep belief networks (Hinton et al., 2006).

[0063] Similarly, Spratling (2008, 2009, 2012) has introduced several recognition systems built of stackable "cortical" modules. These models are composed of modules that work hierarchically and perform a process called "predictive coding", that looks very akin to matching in an ART system. A close examination of the derivation of the learning laws in these systems (Spratling et al., 2009) reveals that they were developed as an incremental version of a well-known batch coding algorithm, non-negative matrix factorization (NMF), developed by Lee and Seung (1997, 1999). The algorithm presented by Spratling at al. does allow incremental (fast) learning, but does not include methods for object segregation/segmentation, scene recognition, and active vision.

[0064] However, none of the above-mentioned object recognition algorithms deals with the issues of how objects are separated from their background, and neither of these models uses space-variant sampling.

[0065] The ARTScan (Fazl et al., 2009) model, the saccading restricted Boltzmann machine (sRBM) (Larochelle & Hinton, 2012), and the entropy minimization algorithm of saccades (Friston et al., 2012)

[0066] The saccading restricted Boltzmann machine (Larochelle and Hinton, 2012) uses space variant vision. However, it does not include a mechanism that informs the system when the system stops fixation from an object and starts fixating on another, which is provided by a human supervisor. The system could not tell apart two identical objects presented side-by-side with a spatial gap separating them.

[0067] The entropy minimization algorithm of saccades (Friston et al., 2012) includes bi-directional *What-to-Where* stream interactions but does not use space-variant vision, and suffers from the same issue as Larochelle and Hinton (2012) in terms of object fixation memory.

NVIDIA Ex1002 Page 319 of 1081

[0068] The ARTScan (Fazl et al., 2009) model includes *Where-to-What* interaction in guiding when the *What* system should learn/stop learning, but does not include *What-to-Where* interactions to inform eye movement and visual search. Additionally, OpenEye differs from ARTScan in these additional dimensions:

- OpenEye and ARTScan use a different log-polar sampling;
- OpenEye shroud formation is feed-forward;
- OpenEye is designed to operate in 3D environment in a noisy background;
- OpenEye is designed to handle self-motion;
- OpenEye employs a concept of temporal continuity to support dynamic scenes;
- OpenEye can combine multiple saliencies, endogenous spatial attention, attention to specific features in order to make next saccade; and
- OpenEye was tested standard MNIST database, whereas ARTScan was tested on handcrafted images.

[0069] Object learning models from Baloch and Waxman (1991), Bradski and Grossberg, (1995), Seibert and Waxman (1992) do use space-variant transformation, or "cortical magnification", but only focus statically at an object's center-of-mass.

[0070] OpenEye methods discussed below employ a learning scheme that maximizes memory efficiency in terms of learning accuracy and capacity to enable both fast and slow stable learning of sensory features.

[0071] **Benefits and Applications**

[0072] Benefits of these methods and systems disclosed herein include providing a single process for identifying, learning, localizing, and tracking objects in an arbitrary sensory system (e.g., data streams derived from static/pan-tilt cameras, cameras, LIDAR, IR, RADAR, microphones arrays, or other networks of sensors, including sensors on one or more mobile robots) and for learning the identity of different sensory scenes. Exemplary embodiments allow quick and stable learning of new patterns without the need to retrain the system, while reducing network (system) size and communication between system components with respect to competing models. The technology disclosed herein is useful to allow continuous learning of arbitrary sensory representations in hierarchies of rate-based or spike-based neural processing stages connected by adaptive (learnable) synaptic weights.

NVIDIA Ex1002 Page 320 of 1081

This technology disclosed herein is general enough to be applicable to any sensory system, and the learning techniques can be applied to two or multiple-stages network, where a neural stage can be a sensory stage and another neural stage can be a higher-order (e.g., categorization) stage. Additionally, the techniquescan be applied to higher-order processing stages, e.g., in higher-order processing stages where representations are more abstract than the one pertaining neural stages at the sensor stage. Additionally, a benefit of this technology is to allow fast learning of new stimuli without the need to interrupt the functioning of the machine. This allows a robot, a camera, a microphone, or another sensor (e.g., LIDAR, RADAR, IR sensor) to quickly learn the identity of a new, previously unlearned input without the need to retrain previously seen input.

[0073] The technology presented herein has applications in designing software to either extract information or control mobile robots, cameras, microphones, motorized vehicles (e.g., self-driving cars) or other networks of sensors. In particular, the technology disclosed herein allows these machines to increase their knowledge base (e.g., the number of visual, acoustic, or other sensors objet it can recognize) over time without the need to retrain the system on the entire knowledge base.

[0074] **OpenEye overview**

[0075] In its first instantiation of OpenSense as a visual system operating on visual data, the OpenEye model is comprised of four main modules: the *Environment* Module, the *Where* system, the *What* system, and an external module that can provide a teaching signal to the what system. These four components are discussed in detail below and shown in FIGS. 1–4.

[0076] The *Environment Module* (e.g., camera 100, microphones 500, and/or wireless sensors 600) abstracts interactions between the vision system and the environment, which can be a virtual environment or a real environment sampled by a fix/pan-tilt camera, a robot-mounted camera, or other visual or non-visual sensory system. This module delivers a visual image to the visual system and executes camera movement commands, which emulate human eye movements. The environment module allows OpenEye to interact with the environment: virtual or real, static or dynamic, real time or prerecorded.

[0077] One task of the *Where System* (130) is to decide where the sensory system should "look" based on salient image properties extracted from the visual image, or based on information coming from the *What System* pertaining to the identity of objects in the environments, and/or the scene identity as a whole. Processing of visual image by the where

NVIDIA Ex1002 Page 321 of 1081

system module includes aspects of the mammalian lateral geniculate nucleus (LGN), primary visual cortex (V1), and higher cortices (V2, MT, MST) processing. The image obtained from the environment module in retinal coordinates, undergoes log-polar transformation to simulate space-variant sampling of the visual input and extraction of features such as (but not limited to) edge, contour, color, and luminance. OpenEye's functioning is not limited to log-polar sampling, and can operate with other space-variant transformations, such as the reciprocal-wedge transform (Tong and Li, 1995), or the pyramid method (Adelson et al., 1984), as examples.

[0078] Also known as the dorsal stream in vision literature (Mishkin and Ungerleider 1982; Webster et al., 1994), OpenEye's *Where System* generates camera movements in order to sample an image by foveation on the spatial location it selects as the most salient, where saliency can be determined by sensory input or semantic (*What System*) information. Foveation is achieved by centering the sensor in the object of interest, so that the object is likely to fall in the center of the space-variant representation. A form-fitting attentional shroud (namely a signal that fits the form, or shape, of an object, similarly to the way a shroud or veil fits the surface it rests on) is then formed around the foveated object. The shroud serves to suppress surrounding objects in order to isolate the object of interest for learning in the *What System*, and enables the system to trigger further camera movements centered exclusively on this enshrouded object. The ability of the *Where System* to form this attentional shroud around a single object has the added benefit of detecting when a foveation has left the previous object of interest. This change in foveated object produces a reset signal that represents temporal discontinuity between the foveations and is used by the *What System* to regulate learning, with the result of allowing OpenEye to group multiple views of an object (but not other objects, or the background) into coherent object categories. Another function of the *Where System* is to maintain a visual working memory of previously foveated locations such that the camera does not persistently choose the same point of fixation. Together with the *Environment Module*, the *Where System* forms the *Where Pathway (140)* that concerns with spatial interaction with the environment and spatial processing.

[0079] The *What System* (150) includes a hierarchy of classifiers that collectively learn to visually recognize an arbitrary number of objects regardless of each object's position and orientation relative to the sensor(s), e.g., a camera. The *What System* receives an object's feature representation as input from the *Where System*. Views are then clustered in an incremental, unsupervised fashion into object representations based either on their similarity

NVIDIA Ex1002 Page 322 of 1081

or according to their temporal continuity as signaled by the *Where System*. The *Where System* provides a shroud-based reset signal, discussed later, that informs the *What System* when seemingly different views are part of the same or different object; this signal is important to OpenEye's ability to learn pose-invariant object representations (Fazl et al., 2009). An optional external *Teacher* (160) provides a supervised learning environment that not only improves classification accuracy and learning speed but also dynamically creates a user-friendly search interface to the visual system's learned knowledge. Because of the hierarchical separation of unsupervised view learning and supervised object-label learning, the *What System* can be switched between unsupervised and supervised learning modes at any time.

[0080] The *What system* and *Teacher* together form the *What Pathway* (170), modeled upon the ventral visual processing stream in the mammalian brain, which concerns the identity of those objects viewed by OpenEye.

[0081] Encoding OpenEye Activity

[0082] One task for OpenEye operation is switching between the coordinate systems centered on the on the robot/camera/sensor (ego-centric), the environment (image-centric or world-centric), and between metrics systems (Cartesian or log-polar). For example, the image can be sampled using a retinal metric (e.g., log-polar) or other metric (e.g., pyramid or reciprocal-wedge), but the signal for the camera to move and how much to adjust the pitch and/or yaw is provided in a Cartesian metric (linear). One role of the *Where System* concerns translating between representations of a signal to different coordinate bases.

[0083] For clarity, the coordinate systems is defined with a term that refers to where the system is centered followed by a term that defines the distance metric of the reference frame. Reference frames can be centered at three possible locations: 1) sensor-centered, 2) ego-centered, and 3) image-centered. Sensor-centered refers to a coordinate system where the (0, 0) location resides at the position of the current camera center. Ego-centered refers to a coordinate system where (0, 0) corresponds to a neutral position of a sensor, with respect which the camera center may be shifted or rotated. Image-centered refers to a reference frame in which the (0, 0) location is at the image center. Image-centered can also be interpreted as global coordinates or scene-centered when the scene is dynamically changing. Correspondingly there are three set of dimensions used in OpenEye: Image Dimensions [W_i, H_i], Sensor Movement Range [W_e,H_e], and Sensor Dimensions [W_s H_s] that represent log-

NVIDIA Ex1002 Page 323 of 1081

polar transform of the Sensor Movement Range. This notation is used in OpenEye description below.

[0084] There are two distance metrics in the coordinate frames: 1) log-polar, and 2) Cartesian. The log-polar distance metric reflects how the eye naturally samples the image and image representation in primary visual cortex, and is employed in the described system by performing a space-variant (log-polar in this case, but other methods could be used) transformation to the ray input, while the Cartesian distance metric is more pertinent when mapping representations onto the real word or for invoking linear control of the eye/camera. In the figures and text below, coordinate frame are referred to as a combination of where it is centered and what defines its distance.

[0085] FIGS. 2A–2C depicts aspects of the *What* and *Where* systems shown in FIG. 1 for an OpenSense architecture that processes visual data (aka an OpenEye system). FIG. 2A shows the Environment Module (120) and the *Where System* (130), which collectively constitute the *Where* Pathway (140). The environment module 120 includes an RGB image sensor 100, which may acquire still and/or video images, whose field of view can be shifted, narrowed, and/or expanded with one or more actuators 110, including but not limited to zoom lenses, tip/tilt stages, translation stages, etc. The environment module 120 provides both image data from the image sensor 100 and actuation data (sensor position data) from the actuator(s) 110 to the *Where* system 130, which in turn provides processed image data to the *What* system 150. The environment module 120 also provides actuation data (sensor position data) from the actuator(s) 110 to the Teacher 160, which forms part of the *What* pathway 170 with the *What* system 150.

[0086] FIG. 2B shows the *Where* system 130 in greater detail. A first log-polar transformation block 260 in the *Where* system 130 performs a log-polar transformation on the image data from the image sensor 100 as described in greater detail below. A feature extraction block 240 identifies features in the transformed image data, which is segmented into bounded regions by a segmentation block 180. A figure/segragation block 210 segregates the bounded regions to form a spatial shroud that fits the foveated region of the image. The figure/segregation block 210 provides a representation of this spatial shroud to the *What* system 150.

[0087] FIG. 2B also shows that the actuator(s) 100 provide sensor position data to a foveation memory 250 and an inhibition of return block 220, which together prevent the

NVIDIA Ex1002 Page 324 of 1081

image sensor from foveating the same portions of the scene (acquiring and/or processing imagery of the same portions of the scene, e.g., at enhanced resolution) unnecessarily. A second log-polar transformation block 230 performs a log-polar transformation on the output of the inhibition of return block and passes the transformed output to a hot spot selection block 190, which determines the next portion of the scene for foveation. A reverse log-polar transformation block 270 transforms the output vector into the frame of reference used by the actuator(s) 100 and provides the transformed output vector to the actuator(s) 100 for actuation of the sensor 100. A temporal object continuity block 200 processes another copy of the hot spot selection block output to determine if the next foveation location falls off the current object surface. If so, the temporal object continuity block 200 transmits a "reset" signal to the *What* system 150

[0088] FIG. 2C shows the *What* system 150 in greater detail. The *What* system 150 uses data from the temporal object continuity block 200, the feature extraction block 240, and the figure/ground segregation block 210 to identify and locate objects in the scene imaged by the image sensor 100. A view layer 280 uses features and shroud data from the *Where* system 130 to cluster shroud-gated visual representations of object views according to their feature similarity. A disambiguation map block 310 generates a disambiguation map of the scene based on these representations from the view layer 280.

[0089] The object layer 290 uses the representations from the view layer 280 to learn pose-invariant object representations by associating different view prototypes from the view layer 280 according to their temporal continuity provided by the reset signal from the *Where* system 130. This yields an identity confidence measure, which can be fed into a name layer 300 that groups different objects under the same user label, which may be obtained from an optional teacher 160. The optional teacher 160 shapes the association between objects and their labels and feeds this information from the Name layer 300 to the Object layer 290 and View layer 280 to the speed and accuracy of future object learning.

[0090] FIGS. 3A–3D provide an overview of how the OpenEye determines temporal object continuity. In block 602, an image sensor, which may or may not be mounted to the robot, obtains imagery of the robot's environment. One or more OpenEye processors translate one or more these images from the camera frame of reference to an allocentric frame of reference (e.g., a log-polar frame of reference) in block 604. The OpenEye processor then segments the translated images in block 606. Next, the OpenEye processor constructs a spatial shroud for a first image (block 608) based on the current position and orientation of

NVIDIA Ex1002 Page 325 of 1081

the input sensor and uses the shroud to identify an object in the first image (block 610). It then translates, rotates, skews, and/or otherwise transforms the shroud to account for the sensor's change in orientation and/or position between acquisition of the first image and a second image (block 612).

[0091] The processor then determines if the transformed shroud maps to an object in the second image (block 614). If so, the processor determines that the object in the second image is the same as the object that appears in the first image and learns the object's location (e.g., stores a representation of the object, its features, and/or its position in memory for later retrieval). At this point, the processor may use an actuator to orient and/or position the sensor in order to image a different portion of the robot's environment. If the shroud does not overlap with an object sufficiently in the second image, the processor determines that the objects are different and updates its memory accordingly. The processor may then actuate the sensor to obtain additional images of the object and the surrounding portion of the robot's environment.

[0092] FIGS. 3B–3D illustrate the shroud construction and translation process. In FIG. 3B, the sensor is centered on a face 702, where the center is marked by the dashed lines through the field of view. The OpenEye processor shroud 704 is built around this face 702, shown by the gray shading in the diagram. After the sensor is reoriented and another image acquired, the shroud 704 is translated and rotated to compensate for the sensor motion. If the sensor is now centered on a location marked by the shroud 704 in FIG. 3C, the system identifies that this object is the same as the one previously viewed. If the sensor is instead centered on a location off of the shroud 704, as in FIG. 3D, the system identifies and learns views of a new object.

[0093] Note that the *What* system (aka the classifier or semantics module) can also contribute to controlling the *Where* system (aka the spatial attention module). In particular, if the *What* system has gathered enough evidence (namely, a certain number of classifications where confidence is high) about the foveated object, it may cause the *Where* system to stop foveating that object, producing Inhibition Of Return (IOR) for a few time steps in the future, so as to bias the visual system to classify other objects in the scene.

[0094] **Implementations of the *What* and *Where* Systems**

[0095] The *What* system (spatial attention module) and the *Where* system (semantics module) can be implemented in hardware, firmware, software, or a suitable combination

NVIDIA Ex1002 Page 326 of 1081

thereof. For example, the *What* and *Where* systems may be implemented as processor-implementable instructions that are stored in non-transient form in one or more memories located in or on a robot, such as a unmanned aerial, ground, or submersible vehicle. Some or all of the processor-implementable instructions may also be stored on remote memory, such memory in or accessible by a server that communicates with the robot via a wireless communication link (e.g., a radio-frequency or optical link).

[0096] The robot may include one or more processors that are coupled to the memory and configured to execute the instructions so as to implement the *What* and *Where* systems, including the individual modules shown in FIGS. 1 and 2A–2C. For example, the robot may execute the instructions with a central processing unit (CPU) and a graphics processing unit (GPU), e.g,. as disclosed in U.S. Patent No. 8,648,867, which is incorporated herein by reference in its entirety. The processor(s) can also be implemented as application specific integrated circuits (ASICs), field-programmable gate arrays (FPGAs), and/or other device or component as understood in the art.

[0097] In some embodiments, some or all of the processors may be located remotely—that is, not on or in the robot. For example, the processors (include GPUs) may be located in one or more smart phones, tablets, and/or single board computers (SBCs). The processors may also form part or all of a cluster computing environment, with each processor in the cluster dedicated to particular task or group of tasks. In these embodiments, the processors may communicate with sensors, actuators, and other devices and components on or in the robot via a suitable communications link, such as a radio-frequency or optical communications link.

[0098] FIG. 5 illustrates an OpenEye system 500 used to control a wheeled robot 510. The OpenEye system 500 includes a computing device 504, such as a tablet computer or other electronic device with wireless capabilities, that is controlled by a user 502. The computing device 504 communicates with the robot 510, which includes an image sensor 512 and an antenna 514, via a wireless link. The user 502 issues commands to the robot 510 via software running on the computing device 504, a processor (not shown) on the robot 510, and/or on other cloud-based processors (not shown).

[0099] In operation, the image sensor 512 can be oriented and/or positioned either by the user when manually operating the robot or automatically by the software. For example, the image sensor 512 may be mounted on a pan/tilt stage, translation stage, or rotation stage that

NVIDIA Ex1002 Page 327 of 1081

can be actuated to change the image sensor's orientation and/or position. The image sensor 512 may also have a (motorized) zoom lens that can be used to zoom in or out on certain portions of the environment. In addition, or instead, the image sensor 512 can be oriented or positioned as desired by moving the robot 510. In some cases, the image sensor 512 may static with respect to the robot 510; this is roughly equivalent to somebody without, say, neck and eye muscles. In order to change the static image sensor's point of view, the body of the robot rotates and/or moves, e.g., using wheels or legs for ground robots, propellers for drones, thrusters for submersible robots, etc.

[0100] Environment Module (120)

[0101] This Environment Module abstracts away the source of visual imagery (cameras, real or virtual, or other sensors, e.g. LIDAR) and applies sensor movement commands in the manner consistent with the environment in which OpenEye currently operates. OpenEye supports the following environments:

Static Scenes – e.g., JPEG, PNG images, etc.;

Dynamic Scenes – e.g., movie files (.avi, .mp4, etc.);

Camera - Real 3d visual world; and/or

Virtual Camera – virtual environment, e.g., based on the JMonkey game engine.

[0102] Concrete implementations of this module are specific to the environment, but the input and the output should comply with the specification below.

[0103] RGB Sensor (100)

[0104] RGB delivers the RGB image sampled from the environment as directed by the RGB Sensor Actuator. The later simulates eye movement by moving the camera.

[0105] Sensor Movement Actuator (110)

[0106] Sensor Movement Actuator implements sensor (e.g., camera) movement commands if they are supported by the environment, otherwise this module returns eye position in ego-centric coordinates.

[0107] *Where* System (130)

[0108] FIGS. 2A and 2B illustrate functions of the *Where System*, including producing a foveated view of the object to be interpreted by the *What System*, to select the next location to foveate based on sensory and internal semantic information, and to determine and track the

NVIDIA Ex1002 Page 328 of 1081

position of objects in the visual field and return their coordinates. The diagram of the *Where System* is presented on FIG. 2B. All modules part of the *Where System* are enclosed in the module described in (130). The *Where System* receives the video image from the environment module and produces camera movement commands to be executed by the environment module (120). The *Where System* supplies the *What System* with the view of the object it currently looks at and the Reset signal, which marks the beginning of the object foveation sequence. The detailed description of the *Where System* modules is presented below.

[0109] Log-Polar transformations

[0110] Several modules (230, 260, 270) perform transformation between log-polar and Cartesian encoding of the distance metric. OpenEye adheres to a bio-inspired log-polar transform of the input image, but the model can be used with different transform. The log-polar transform is applied to the RGB sensor subtending 136 degrees of visual angle, close to that reported in humans and other animals (Traver and Bernardino, 2010). The log-polar metric in space encoding is used across both OpenEye *Where* and *What Systems* and transformed back to the Cartesian metric by (230) only to reference the external world beyond the current view, which is required by the Environment Module (120), the Foveation Memory module (250), and the Inhibition of Return module (220) in the *Where Pathway*. All Log-polar transformation modules share the parameters that specify dimensions of log-polar $[w_s h_s]$ and Cartesian image $[W_s H_s]$.

[0111] Log-Polar transformation of retinal RGB image (260)

[0112] The image sampled at the foveated location undergoes log-polar transformation that amounts to space-variant sampling with higher resolution in the foveal area and much coarser resolution that falls with eccentricity outside the foveal region (Traver and Bernardino, 2010). This provides some invariance to translation/rotation and to save computational bandwidth while at the same time to acquire details at the location of the image that present the highest interest and is the most effective for the image representation.

[0113] Log-Polar transformation of inhibition of return (230)

[0114] Similarly to retinal image, inhibition of return undergoes log-polar transformation in order to prevent the HotSpot Selection Module (190) from repeated foveations.

[0115] Reverse Log-Polar transformation of inhibition of return (270)

NVIDIA Ex1002 Page 329 of 1081

[0116] The HotSpot selected in the Log-polar view in sensor-centered coordinates (190) needs to be transformed back to Cartesian metric by (230) before it can be converted into sensor movement command by the Environment Sensor Movement Actuator (110).

[0117] **Feature Extraction (240)**

[0118] Feature Extraction (240) includes, but is not limited to, computation of luminance and color. Other features could include motion, or SIFT features (Lowe, 2004). "Features" can include, but are not limited to:

A property of an image that can be associated with each image location;

A scalar (e.g., luminance, 0-dimensions) or vector (e.g., color, 1 dimension);

A numerical (integer, or real, e.g. luminance, color) or binary (Boolean, e.g., is an edge associated with this particular pixel) value. More abstract properties (e.g., "edgeness") can also be represented by a numerical feature – strength of the edge.

[0119] The description below specifies the features currently implemented in OpenEye, but the description below should not be intended to limit OpenEye applicability to these features alone.

[0120] **Luminance, Color**

[0121] Luminance and Color can be extracted from the Log-polar RGB Image.

[0122] **Segmentation (180)**

[0123] This module builds preliminary segmentation producing binary image that represent closed (bounded) regions (Suzuki & Abe, 1985). This is achieved by using OpenCV function ***findContours***, which operates on edges produced by the Canny edge detector (Canny, 1986). The result is the image with pixels set to 1 at the locations that belong to the bounded regions.

[0124] **Figure/ground segregation (180)**

[0125] This module builds a shroud around the object at the center of the view. This is achieved via a seeded flood fill algorithm, which uses the OpenCV **floodFill** function. This algorithm fills a connected component starting from the center of the log-polar image produced by the segmentation module (180). Connectivity is determined by the brightness closeness of the neighbor pixels. As the result this step produces a shroud (Fazl et. all, 2009).

NVIDIA Ex1002 Page 330 of 1081

roughly fitting the form of the closed region that includes the foveated location (the center of the image).

[0126] The Figure/ground segregation module (180) can also be extended to accept input from the *What System*, for instance in the form of semantic information pertaining the identity of pixels which can be obtained via a fast processing of the visual information that bypasses the *Where System*. For instance, a separate *What System* can be trained to recognize, on a pixel-by-pixel basis, areas in the image. E.g., the separate *What* System can initially classify areas of the image as "sky", "grass", "road", and this information can be used as input to the Figure/ground segregation module (180) as additional input to drive figure/ground segregation.

[0127] **Hot Spot Selection (190)**

[0128] This module produces a vector that determines the next foveation location. The module determines the most salient locations on the image by using the OpenCV function **goodFeaturesToTrack**, which finds the most prominent corners in the image as described in (Shi and Tomasi, 1994). The image passed to the corner finding algorithm is the luminance feature produced by the feature extraction module (240). The Inhibition of Return signal produced by the log-polar transformation module (230) disables the non-zero locations on the image to be selected as the next foveation position.

[0129] **Temporal object continuity (200)**

[0130] In order to build view invariant object identity, OpenEye may maintain temporal continuity between subsequent object foveations. OpenEye determines if the next foveation location falls off the current object surface in order to signal the object recognition system that building of the new object identity begins or continues. This is achieved via producing the RESET signal, which is *set to 1* in the next cycle when the selected new foveation location falls off the shroud (output of module 210) built from seeded activity in the center point of the view.

[0131] In OpenEye, temporal continuity is based on the ability to learn the location of the object selected during the foveation (camera movement) cycle. The location is learned by translating pixel position corresponding to the object in the camera-centered coordinates into object location in allocentric coordinates.

NVIDIA Ex1002 Page 331 of 1081

[0132] To ensure awareness of previously learned objects, their locations are translated from allocentric coordinates stored in object memory into camera-centered representation at each foveation cycle.

[0133] Similarly to objects, hot spot pixel position is translated to allocentric coordinates. In the next foveation cycle, the position of hotspot is recomputed forming the shroud around the foveated object by seeded filling-in starting from the hotspot selected at the previous foveation cycle.

[0134] **Foveation Memory (250)**

[0135] The term "foveation" adopted below is borrowed from the neuroscience literature, where foveation represents the location of eye fixation. Foveation memory in OpenEye represents past foveation activity over the visual image. When OpenEye operates on static images, foveation means sampling of the image, at a particular (foveated) location. Usually size of the sampled image is much smaller than the entire image(scene) size. When OpenEye operates in real 3D or virtual environment, foveation is sampling of that environment as the result of real or virtual camera movement. The visual memory is maintained over the spatial area that depends on the environment. It could amount to the entire image as in the case of static scene environment, or over the region of space that is currently in the view as in the case of movies or virtual environment. Foveation memory inhibits foveations at the locations that have been foveated in the past. After making a camera movement, OpenEye sets foveation activity at the maximum value (255), this activity decays with each foveation and eventually, when it decays to 0, the location is enabled for new foveations. The Foveation Memory is maintained in the image-centered coordinate frame. However, the input (Sensor Position, 150) is provided in ego-centered coordinates.

[0136] The history gets updated with each new foveation cycles. The decay is implemented as a decrement by one with each foveation step. Initial value immediately after foveation is set to FMAX. This means that the same location cannot be foveated at least the next FMAX cycles.

[0137] **Inhibition of Return (220)**

[0138] The purpose of the Inhibition of Return module (220) is to prevent repeated foveations at the same spatial location. To achieve that this module extracts the section of the foveation history around the next foveation location that falls in the view of the next saccade.

[0139] Input:

NVIDIA Ex1002 Page 332 of 1081

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Metric	Base
Sensor Position $i_s=[x,y]$	Vector	0	Unsigned, Byte	2	Cartesian	Ego-centered
Foveation Memory I_f	Scalar Field	$[W_f H_f]$	Unsigned Byte	0	Cartesian	Image-centered

[0140] Output:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Metric	Base
Inhibition of return O	Scalar Field	$[W_s H_s]$	Unsigned Byte	0	Cartesian	Sensor-centered

[0141] Processing:

$$\begin{cases} I_f(X,Y) & 0 < X < W_f;\ 0 < Y < H_f \\ 0 & X > 0 \mid X > W_f \mid Y < 0 \mid Y > H_f \end{cases}$$

where

$$Y = y + i - i_0;\quad X = x + j - j_0;$$

$$i_0 = \frac{(H_s-1)}{2};\ j_0 = \frac{(W_s-1)}{2}.$$

[0142] ***What* System (150)**

[0143] As shown in FIGS. 2 – 4, the *What System* (150) learns the identities of objects and visual scenes. The *What System* may, for example, group object views and learn them as a unified entity; maintain a lifelong memory while preventing memory degradation and saturation; make inferences about views acquired from the *Where System*, which objects they belong to, and the names associated with those objects; bias camera movements in the *Where System* in order to intelligently guide image sampling; and provide an interface by which an external user can communicate with and leverage the system's knowledge.

[0144] The *What System* is implemented as a series of hierarchically organized classifiers that perform unsupervised *view* clustering, classification of view categories into *object* categories based on the reset signal from the *Where System*, and supervised or unsupervised categorization of objects into *name* categories. After learning occurs, the activation of a name

NVIDIA Ex1002 Page 333 of 1081

category primes the *What system* by inhibiting those objects and views that are not associated with that name category, further tuning the system by discouraging views from being shared between multiple objects and names. The activation of a name category can come from a bottom-up activation of a newly viewed object, persistent activation caused by a previously viewed object, or through external activation by another system or user. This external activation is provided by a Teacher (160) that represents the correct name of the foveated object to aid learning. OpenEye does not function in different modes to facilitate training or testing mechanisms, and it does not require a reset of the system upon transition to a new scene. FIG. 2C highlights the high level system diagram of the *What Pathway* (170), which includes the *What System* (150) and the Teacher (160).

[0145] The inspiration for hierarchical clustering of views into objects and names is detailed in the ARTScan model of visual learning (Fazl, Grossberg, and Mingolla, 2009). The Adaptive Resonance Theory (ART) learning scheme has been altered from this work by replacing the learning system of the view layer with a variant of Fuzzy Simplified ART (f-sART; Baraldi and Alpaydin, 1998); differences between the OpenEye view layer and f-sART are detailed in Section 4.1. Additionally, the specific mechanisms and learning rules implemented in the object and name layers have been altered in order to enhance learning quality and to allow the system to operate with or without an external teacher; these differences are described in their respective sections.

[0146] The following sections describe the function of the *What* Pathway, shown in FIG. 2C, in detail. The View layer (280), described herein, clusters shroud-gated visual representations of object views according to their feature similarity. The Object layer (290), described herein, learns pose-invariant object representations by associating different view prototypes according to their temporal continuity provided by the reset signal from the *Where* system. The Name layer (300), described herein, further groups different objects under the same user label if given from an optionally present Teacher (160), described herein. As an external teacher shapes the association between objects and their labels, this information is fed back from the Name layer to the Object and View layers to improve the speed and accuracy of future object learning.

[0147] **View layer (280)**

[0148] The first task of the *What pathway* is to cluster the shroud-determined input surface properties generated from the *Where System* into consistent view categories, which is

NVIDIA Ex1002 Page 334 of 1081

performed by the *View layer* (280). This layer learns a library of typical views of a set of objects in different poses and spatial configurations: a set of views connected to the same object category node, described herein, should correspond to a set of 2D feature views of an object that together represent the view of this object from varying 3D angles, distances, lighting conditions, and other variations in viewing experienced by the OpenEye system. While the log-polar representation provides some form of invariance to scale and translation, this learning mechanism is at the core of how OpenEye learns invariant object and scene representations. The features received by the view layer comprise a set of m different analog values associated with each of a set of pixels. The value of the shroud at that pixel determines whether those m features will be part of the comparison that determines which object view most closely matches the currently viewed object. These features can include luminance or color information, the presence and orientation of local contours, local motion information denoted by optic flow, stereo disparity, binary feature descriptors such ORB representations (Rublee, Rabaud, Konolige, & Bradski, 2011), or any combination thereof that can be produced by the *Where system*.

[0149] The unsupervised learning network that does this clustering is a simplified fuzzy Adaptive Resonance Theory (f-sART) network (Baraldi & Parmiggian, 1997; Baraldi & Alpaydin, 1998; Baraldi & Alpaydin, 2002). A thorough description of adaptive resonance theory is presented elsewhere (Carpenter & Grossberg, 1987; Carpenter & Grossberg, & Rosen, 1991; Carpenter & Grossberg, 1995). ART clarifies how matching between bottom-up input and top-down representations enables fast and stable learning of arbitrary input patterns. Most recently, ART has been shown to be biologically plausible at the level of laminar multi-compartment spiking neurons, and consistent with experimental data (synchronous matching ART; Grossberg & Versace, 2008)—a task that competing models such as hierarchical temporal memories have not yet achieved (George & Hawkins, 2009).

[0150] Simplified fuzzy ART, or f-sART, departs from the classic fuzzy ART formulation in three ways:

1) The category activation function and presentation-to-category matching function are both bidirectional and symmetric. This symmetry simplifies the process of finding a resonant neuron to direct calculation rather than an iterative ART search cycle.

2) Fuzzy sART imposes soft-competitive learning rules whereby a view can activate a resonant domain instead of a single resonant view neuron, allowing non-winning weights to

NVIDIA Ex1002 Page 335 of 1081

be updated as well as the winning category weights. The soft-competitive learning rules of f-sART share properties of Kohonen's self-organizing maps, which have the added advantage of autonomously determining sparsity at the view category layer.

3) Fuzzy ART is not a *consistent* statistical learning system in that its learned category clusters depend on the order of sample presentation. Fuzzy sART partially overcomes this limitation by training in repeated batches and removing redundant or infrequently used categories between training batches.

[0151] OpenEye leverages improvements (1) and (2) for fast category matching and the creation of robust distributed object representations. Because OpenEye learns online and without forced repetition, however, and the dataset size is unknown beforehand and depends on the pattern of *Where system* foveations, OpenEye can leverage a novel category consolidation technique described herein.

[0152] Unlike many unsupervised learning systems, f-sART only takes two user defined parameters, vigilance (a parameter that determines how close of a match is close enough for resonance and learning to occur) and τ (a parameter that mediates how many activations of the same category node are required for it to become a stable representation for the category). This mitigates the problem of parameter tuning and offloads as much of the work as possible to autonomous mechanisms that self-tune sparsity of the view category code. These properties balance biological plausibility, simplicity, and accuracy in a way that make f-sART a practical OpenEye view clustering system.

[0153] The View layer (280) functionality can be broken down into three sub-processes described below:

1) determining the winning view category neuron and its resonant domain;

2) updating the state of the view category nodes in the resonant domain; and

3) producing a disambiguation map that can prime the *Where System* for new foveation positions that can efficiently determine the currently viewed object's identity.

[0154] A summary of the input-output relationships of the View layer is given in the following table.

[0155] Input:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation

NVIDIA Ex1002 Page 336 of 1081

Features	Scalar Field	$[w_s, h_s]$ (log-polar, sensor-centered)	Unsigned, Byte	m	$\vec{x}$	240 (O_L)
Shroud	Scalar Field	$[w_s, h_s]$ (log-polar, sensor-centered)	Binary, Byte	1	$\vec{A}$	210 (O)
Object category activity	Scalar Field	[j]	Floating Point	1	$\vec{o}^{v}$	280.7a
Name category activity	Scalar Field	[j]	Floating point	1	$\vec{n}^{v}$	280.2a

[0156] Output:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation
View category activity	Scalar Field	[j]	Floating point	1	$\vec{v}$	280.3
Disambiguation map	Scalar Field	$[w_s, h_s]$ (log-polar, sensor-centered)	Floating point	1	$\vec{y}$	280.7

Persistent state:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation
View templates	Vector Field	[j]	Floating point	$[w_s \times h_s \times m]$	$\mathbf{W}^{xy}$	280.6
View template ages	Scalar Field	[j]	Floating point	1	$\vec{t}$	280.4

[0157] **View output and the resonant domain**

[0158] To determine which view category neuron best matches the given view input, we calculate the vector degree match between the input and each view category neuron (Equation 280.1). This match is determined as a Gaussian function of Euclidean distance between input and weights in feature space; the width of this Gaussian is automatically adjusted by the size of the shroud in order to retain sensitivity to different inputs regardless of the feature vector's dimensionality. The view category neuron with the highest vector degree match to the input is considered to be the winning category node. $\vec{W}_j^{xy}$ represents the weight vector connecting the input feature scalar field, $\vec{x}$, to the jth view category output node. The input feature field is comprised of the luminance values of the 2D shroud-modified contour reshaped into one long vector, while $\vec{W}_j^{xy}$ can be thought of as the jth data element of the $\mathbf{W}^{xy}$

NVIDIA Ex1002 Page 337 of 1081

vector field. The vector degree match M_j of the input $\vec{x}$ to a particular view prototype j is given by

$$M_j(\vec{x},\vec{A},\mathbf{W}^{xv}) = \exp\left(\frac{-\left(\hat{A}\otimes(\vec{x}-\vec{W}_j^{xv})\right)^{\top}\left(\hat{A}\otimes(\vec{x}-\vec{W}_j^{xv})\right)}{\hat{A}^{\top}\hat{A}}\right), \quad (280.1)$$

$$B_j = \begin{cases}\max(M_j-\rho^v,0) & n_j^v > 0.2 \text{ and } j=h\in\{1,2,\ldots,9\}\\ 0 & \text{otherwise}\end{cases}, \quad (280.2)$$

[0159] where a cell's resonant domain value B_j is above zero if the match exceeds the user-set vigilance parameter ρ^v, the view is associated with the currently active top-down priming signal from the name layer n_j^v, and the cell wins a lateral competition implemented as a ranking denoted by the index h. More specifically, top-down priming from the name to view layers is defined as

$$\vec{n}^v = \begin{cases} f_\infty(\mathbf{W}^{vo}\mathbf{W}^{on}\vec{n}) & R\vee T\\ \vec{1} & \text{otherwise}\end{cases}, \quad (280.2a)$$

[0160] where $\mathbf{W}^{vo}$ are the weights from view to object cells, $\mathbf{W}^{on}$ are the weights from object to name layers, $\vec{n}$ is the output of the name layer described herein, $R\vee T$ is true either when there is temporal continuity defined as a lack of reset signal from the *Where Pathway* (R) or when a teaching signal is present (T), and $f_\infty(\cdot)$ is a normalization function given by

$$f_a(\vec{y}) = \frac{\vec{y}}{\sqrt[a]{\sum_i |y_i|^a}}. \quad (280.2a1)$$

[0161] Note that, since all simulated classifier neuron activities are nonnegative, some of these normalization equations can be programmed in a simplified form, such as $f_1(\vec{y}) = \vec{y}/\sum_i y_i$ and $f_\infty(\vec{y}) = \vec{y}/\max_i y_i$. Cells are also ranked by their match, shown as the index h and described by the ordering

$$M_h \geq M_{h+1}. \quad (280.2b)$$

[0162] The view layer successfully matches the input if the winning node passes vigilance and is associated with the teacher's activated name, given by the condition $B_1 > 0$. If this condition is false, then the view layer attempts to create a new node with a weight vector identical to the currently viewed feature vector. If there are no new nodes available, however, the view layer does not learn, and its output is a vector of all zeros except for that of

NVIDIA Ex1002 Page 338 of 1081

the winning node, whose activity is set to its vector degree match regardless of whether it passes vigilance.

The output of the view layer is a vector of zeros except for those neurons in the resonant domain, whose activities are proportional to their level of match to the input. Except for when match fails and the view layer is out of neurons, the activities are normalized according to

$$\vec{v} = f_\infty(\vec{B}) \quad (280.3)$$

so that the maximal value is always 1.

[0163] View category representation update

[0164] Once the winning category node and its resonant domain are established, the learning rates of these active view category neurons are calculated. The learning rate is a function of neuron membership and neuron match rank, both of which become lower with the neuron age that increases when it is in the resonant domain. The membership function of a neuron is affected both by its match score and by its match rank; the winning node has the highest membership, while the ninth best matching node has a low membership even if its vector degree match score is high.

[0165] The age of a neuron increases with each time it is selected as a winning or resonant neuron that passes resonance, where the winning node ages more quickly than a lower ranked node in the resonant domain. For a given value of the membership function, the learning rate starts high and slowly decreases with neuron age; qualitatively, this symbolizes a stiffening of learning the more times a neuron is activated by a view. The parameter τ controls how fast the learning rate changes as neurons age. Alternatively, one can think of τ as mediating how many activations of the same category node are required for the node to become a stable representation for the category. Since a weaker agreement between input and category layer leads to a more distributed code in the category layer, τ is one of the dominant factors that mediate how quickly a distributed code becomes sparse. τ is one of two user-specified parameters that do not change throughout the simulation.

[0166] ***Neuron age.*** The age of a neuron t_h in the resonant domain is updated after it learns so that it becomes a stable category representation over time. The neuron ages more slowly if it has a lower rank in the resonant domain; this rate is controlled by the equation

$$t_h \leftarrow t_h + \left[n_j^\rho > 0.2\right] \cdot \max(10 - h, 0) \quad , \quad (280.4)$$

NVIDIA Ex1002 Page 339 of 1081

[0167] where [·] denotes an Iverson bracket (1 if the interior condition is true, 0 otherwise) that stops neuron aging if it does not learn from a top-down name mismatch, and the other term controls neuron aging according to its resonant domain rank h.

[0168] ***Weight update.*** The input view prototypes are updated according to a node-depending learning rate that depends on a number of factors. The total learning rate α_j^v for a particular view node is a product of the match (q_j) and rank (s_j) learning rates

$$\alpha_j^v = q_j s_j \quad . \quad (280.5)$$

[0169] The match learning rate is dependent both on the level of feature match to the input and neuron age t_j. This rate is given by the equation

$$q_j = \begin{cases} C_j\left(\varepsilon/C_j\right)^{t_j/\tau} & C_j \geq \varepsilon \\ \varepsilon\left(C_j/\varepsilon\right)^{t_j/\tau} & \text{otherwise} \end{cases} \quad (280.5a)$$

[0170] where $\varepsilon = 0.01$ is small and C_j is a normalized match score:

$$\vec{C} = f_1\left(\vec{B}\right) \quad . \quad (280.5a1)$$

[0171] The rank learning rate is dependent both the cell's resonant domain rank and on its age; when the neuron is young, it is more likely to learn even if its rank in the resonant domain is low. This rate is

$$s_j = \exp\left(\frac{(1-h)}{\sigma_j}\right) \quad , \quad (280.5b)$$

[0172] where the age-based neighborhood spread constant is

$$\sigma_j = 5(0.01/5)^{t_j/\tau} \quad (280.5b1)$$

[0173] and τ is a user-defined time constant that specifies how quickly the network stabilizes. Finally the view layer weights $\mathbf{W}^{xv}$ for each view node j are updated according to the rule

$$\vec{W}_j^{xv} \leftarrow \left(1-\vec{A}\right)\otimes\vec{W}_j^{xv} + \vec{A}\otimes\left(\left(1-\alpha_j^v\right)\vec{W}_j^{xv} + \alpha_j^v\vec{x}\right) \quad , \quad (280.6)$$

[0174] where the weight is mixed with the input $\vec{x}$ in proportion to the match-, rank-, and age-based learning rate α_j^v and is gated by the shroud $\vec{A}$ through element-wise multiplication denoted by the $\otimes$ symbol.

[0175] **Disambiguation map (310)**

NVIDIA Ex1002 Page 340 of 1081

[0176] A single input view passed to the *What System* can activate multiple view, object, and name nodes. Although the output of each of these layers is sparse, the system output can occasionally be unsure about object identity in the absence of an external teacher. This is called "object ambiguity", as a single view of an object can be associated with many objects.

[0177] To facilitate object disambiguation, OpenEye uses a novel, dynamically constructed, *disambiguation map* that suggests potential saccade targets to the *Where Pathway* that would maximally inform the *What System* as to which of the potential object representations best matches the actual viewed object. This map compares those views within the resonant domain that are activating disparate object categories, and activates the disambiguation map in the areas where input and view prototypes disagree.

[0178] The map is currently defined as the weighted average of the feature differences between the input $\vec{x}$ and weight templates $\vec{W}_j^{xv}$, where each template is mixed only if a view v_j in the resonant domain is coactive with its associated object category o_j^v. Specifically, the disambiguation map $\vec{\gamma}$ is defined as

$$\vec{\gamma} = \frac{\sum_j o_j^v v_j \left| \vec{W}_j^{xv} - \vec{x} \right|}{\sum_j o_j^v v_j} \quad , \quad (280.7)$$

[0179] where o_j^v is the feedback activity of the object category layer to a particular view category j, whose activities are given by

$$\vec{o}^v = f_\infty(\mathbf{W}^{vo}\vec{o}) \quad . \quad (280.7a)$$

[0180] Note that equation 280.7 could be modified to include multiple features by including a sum over features m.

[0181] **Object layer (290)**

[0182] Once the View layer (280) in the *What System* produces view categories that are excited by the shroud-modified contour, the view category neurons are then grouped into object categories—the goal being to determine which views should be bound to a consistent object representation. Intuitively, the Object layer (290, FIG. 2C) makes sure that views of the same object, e.g., different poses of a cup, a car, or a person, all map to the same object node that represents the collective views defining that specific cup, car, or person. The inputs to the object category layer include: the view-category output from f-sART; the reset signal from the *Where System*; and feedback input from the name category.

NVIDIA Ex1002 Page 341 of 1081

[0183] The output of the object category layer are the sparse activities of object category neurons that are activated in response to bottom-up view input and top-down name input, produced as a simplified model of a Recurrent Competitive Field (RCF) neural network (Grossberg, 1973). Unlike previous models, the weights from view to object neurons are learned through a modified form of Oja's rule, an associative learning rule that implements favorable input clustering properties such as weight normalization and a distribution of weights that reflects the principle source of variation across a dataset (Oja, 1982).

[0184] The object category functionality can be broken down into two sub-processes:

1) determining the winning object category neurons; and

2) updating the weights between the view category nodes and the winning object category neurons.

[0185] A summary of the input-output relationships of the Object layer is given in the following table.

[0186] Input:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation
View category activity	Scalar Field	[j]	Floating point	1	$\vec{v}^{o}$	290.1a
Name category activity	Scalar Field	[k]	Floating point	1	$\vec{n}^{o}$	290.1b
RESET	Scalar	1	Binary Byte	1	R	200 (O)

[0187] Output:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation
Object category activity	Scalar Field	[k]	Floating point	1	$\vec{o}$	290.2
Identity confidence	Scalar	1	Byte, unsigned	1	c	290.3

[0188] Persistent state:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation
Object categories	Vector Field	[k]	Floating point	[j]	$\mathbf{W}^{vo}$	290.4
Object category	Scalar Field	[k]	Floating point	1	$\vec{u}$	290.1

NVIDIA Ex1002 Page 342 of 1081

internal state						

[0189] Object category output

[0190] The object category layer's output is determined both by feedforward/feedback input and by its previous activation saved in its internal state. This internal state is modeled by a normalized additive neuron equation that approximates the action of an RCF; this layer takes as input: the active view categories based on the current view; and direct feedback from the name layer that primes previously made associations between a current label and a set of candidate object categories.

[0191] The object category winners are defined as those neurons whose view-to-object network weights are more similar to the active view category layer than a fixed vigilance criterion, constrained by top-down feedback signaling whether that view belongs to the current active name category as well. If none of the object category neurons meet this fixed resonance criterion, a new neuron is created as a new object category to be associated with the current view.

[0192] A dominant feature in OpenEye is that the *Where System* should inform the *What System* when the foveated object has changed; until that time, the *What System* should keep grouping views into the same object category. Although the selection of the object category winners happens at each view presentation, without the presence of a reset signal, feedback from the name category layer will keep the object layer locked in its current activation state regardless of the view layer's activity. This allows the *What System* to associate multiple, disparate views of the same surface/object. This persistent state is broken and reset when the *Where Pathway* notifies the object category layer that the current view is of a different object in the scene. In the event of a reset signal, the category layer's activity is set to zero, and top down priming is ignored unless the name layer is strongly activated by an external teaching signal.

[0193] The object category internal state vector $\vec{u}$ is determined by the discrete update equation

$$\vec{u} \leftarrow f_\infty(0.9\vec{u}[R] + (1 - \alpha^o)\vec{v}^o + \alpha^o\vec{n}^o[R \vee T]) \ , \quad (290.1)$$

[0194] where $[R]$ is an Iverson bracket that is 1 in the absence of a reset signal and clears the layer's memory during a reset, $[R \vee T]$ is an Iverson bracket that clears top-down priming during a reset in the absence of a teacher T, $\alpha^o = 0.6$ is a user-defined mixing fraction

NVIDIA Ex1002 Page 343 of 1081

between feedforward and feedback input, $\vec{v}^o$ is the bottom-up input from the view layer given by

$$\vec{v}^o = f_\infty((\mathbf{W}^{vo})^T \vec{v}) \quad , \quad (290.1a)$$

[0195] and $\vec{n}^o$ is the top-down feedback from the name layer given by

$$\vec{n}^o = f_\infty(\mathbf{W}^{no}\vec{n}) \quad . \quad (290.1b)$$

[0196] The output $\vec{o}$ of the object category layer are those cell activities that exceed a user-defined activity threshold $\rho^o = 0.55$; these sparse activations are then normalized by the equation

$$\vec{o} = f_\infty(\max(\vec{u} - \rho^o, 0)) \quad (290.2)$$

[0197] so that the winning object category activity is set to 1.

[0198] Once the *What System* has either successfully recognized the currently viewed object as an instance a previously learned category or learned a new object, the *Where System* is informed of this recognition in order to mark the entire object's extent as unnecessary for further viewing. This recognition signal can be used as a feedback signal to triggers the inhibition of return mechanism, which inhibits the need to sample additional views of the recognized object. The *Identity confidence* of the *What system, c*, is defined as

$$c = 1/\sum_k o_k \quad , \quad (290.3)$$

[0199] which is inversely proportional to the total object layer activity. If only one object category node is active, then $c = 1$, signaling successful recognition so that the *Where system* can add the entire object's spatial extent to the *Inhibition of return* map (Box 220). If multiple objects are simultaneously active, the identity confidence decreases, signaling the *Where system* that the currently viewed object requires more foveations to be unambiguously identified. In this case the *What system* simultaneously provides a *Disambiguation map* (Equation 280.7) that suggests foveation points that may resolve the ambiguity and increase identity confidence.

[0200] **Object category weight update**

[0201] Once the object category winners are selected, learning can occur to associate the active view categories with the selected object category. The object category's learning rule is a modified form of Oja's rule (Oja, 1982) that is equivalent to using Oja's rule to learn the mapping from object to view category layers, which is a novel rule present only in OpenEye.

NVIDIA Ex1002 Page 344 of 1081

This reversal is useful because many views map onto a single category, where most views are inactive at once. Oja's rule is postsynaptically gated, so learning the map from view to object categories would cause the decay of most rarely active view associations with a more frequently active object category. Learning the reverse map with Oja's rule allows this same postsynaptic gating to produce a sparse and consistent one-to-many mapping from object categories to views. The learning rate of this weight update, η, takes on a different value whether the teaching signal at the name layer is present (supervised learning) or absent (unsupervised).

[0202] The weights $\mathbf{W}^{vo}$ between view and object categories are described for each synapse associating view j with object k by the equation

$$W_{jk}^{vo} \leftarrow W_{jk}^{vo} + \eta v_j \left(o_k - W_{jk}^{vo} v_j\right) \quad , \quad (290.4)$$

[0203] where η is a teacher-dependent learning rate. When the teacher is present, $\eta = 0.5$, and when the teacher is absent, these associations are learned at the slower rate of $\eta = 0.05$.

[0204] **Name layer (300)**

[0205] The last hierarchical stage of the *What System* is the name category classifier. The name category network groups different object category neurons with name category neurons using an externally provided teaching signal of the object name that may be present, partially present, or entirely absent.

[0206] The name category functionality can be broken down into two sub-processes:

1) determining the winning name category neurons; and

2) updating the weights between the object category nodes and the winning name category neurons.

[0207] A summary of the input-output relationship of the Name layer is given in the following table.

[0208] Input:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation
Object category	Scalar	[k]	Floating point	1	$\vec{o}^n$	300.1a

NVIDIA Ex1002 Page 345 of 1081

activity	Field					
External Label	String	1				N/A

[0209] Output:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation
Name category activity	Scalar Field	[ℓ]	Floating point	1	$\vec{n}$	300.2
Object label	String	1				N/A

[0210] Persistent state:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation
Name categories	Vector Field	[ℓ]	Floating point	[k]	$\mathbf{W}^{on}$	300.3
Name category internal state	Scalar Field	[ℓ]	Floating point	1	$\vec{p}$	300.1
Label map	String Dictionary	[ℓ]	Floating point	[ℓ]		N/A

[0211] **Name category output**

[0212] The object category layer's internal state is modeled by a normalized additive neuron equation that approximates the action of an RCF; this network takes as input the object category winners; and an optional, external teaching signal with the object name.

[0213] Similarly to the object layer, the bottom-up name category winners are defined as the neuron whose object-to-name network weights are sufficiently similar to the active object category layer to pass a fixed vigilance criterion. A teaching signal unambiguously specifies the name layer (and often object layer) activity values in order to consistently link object

NVIDIA Ex1002 Page 346 of 1081

category representations with string labels that can be used by an end-user to recall or search for particular objects and views.

[0214] The name category output is a function of its internal state vector $\vec{p}$, determined by the discrete update equation

$$\vec{p} \leftarrow f_\infty\left(0.9\vec{p} + (1-\alpha^n)\vec{o}^n + \alpha^n\vec{\Omega}\right) \ , \quad (300.1)$$

[0215] where $\alpha^n = 0.6$ is a user-defined mixing fraction between feedforward and feedback input. $\vec{\Omega}$ is a binary vector from an external teacher, set to all zeros except for the node associated with a particular semantic label (or set to all zeros in the absence of a teacher), and $\vec{o}^n$ is the bottom-up input from the object layer given by

$$\vec{o}^n = f_\infty((\mathbf{W}^{on})^\top \vec{o}) \ . \quad (300.1a)$$

[0216] The name layer can create a top-down priming vector in response to an optional input from an external Teacher (160) because it contains a dictionary that maps string labels to name node indices. Upon receiving a label from the teacher, the name layer checks whether the label already exists in the dictionary; if so, it creates a vector that strongly activates that name layer node. Feedback from the name layer to the object and view layers ensures that learning is then restricted to associating those views and objects that match the current teacher label. If the label has never been seen before, the name layer creates a node and associates the current view and object with that name. After this learning, the name layer can also recall a label by returning the string label associated with the maximally active name category node.

[0217] The output $\vec{o}$ of the name category layer are those cell activities that exceed a user-defined activity threshold $\rho^o = 0.35$; these sparse activations are then normalized by the equation

$$\vec{n} = f_\infty(\max(\vec{p} - \rho^o, 0)) \quad (300.2)$$

[0218] so that the maximal name category layer output value is always 1.

[0219] **Name category representation update**

[0220] Once a name category winner is selected, learning can occur to associate the active object category with the selected name category. Similarly to the object category layer, the name category learning rule is a modified form of Oja's rule that is equivalent to using Oja's rule to learn the mapping from name to object categories (Equation 36) for the same reasons discussed in Section 3.2.2. Similarly to object category learning, the learning rate, η,

NVIDIA Ex1002 Page 347 of 1081

takes on a different value whether the teaching signal at the name layer is present (supervised learning) versus absent (unsupervised).

[0221] The weights $\mathbf{W}^{on}$ between object and name categories are given by a learning rule that is equivalent to Oja's rule when learning the association from name categories to object categories, described for each synapse associating object k with name ℓ by the equation

$$W_{k\ell}^{on} \leftarrow W_{k\ell}^{on} + \eta o_k (n_\ell - W_{k\ell}^{on} o_k) \quad (300.3)$$

[0222] where η is a learning rate whose dependence on the existence of an external teacher is described after Equation 290.3.

[0223] **Teacher (160)**

[0224] The teaching signal, supplied by the Teacher (160), is an optional string label that informs the *What System* about the currently-viewed object's name. The teacher supplies a string label to OpenEye, which automatically assigns that string to a name layer category node. The teaching signal is then transformed by the Name layer (290, FIG. 2C) into a vector whose value is 0 in all locations with the exception of the one corresponding to the object name (whose value is 1). When the teaching signal is present, the top-level classifier of the *What System* (the object-to-name layer) is said to be working in a supervised mode. However, it is important to note that even in supervised mode, the input-to-view and view-to-object layers continue to learn in a largely unsupervised fashion.

[0225] The Teacher is separately implemented for a variety of applications, so no explicit equations are listed for its function. Depending on the application, a teacher can either take the form of categorized class outputs in a standard machine learning database, a string label provided by an external resource such as an Internet image search, or a label provided by a human user, enabling interactive correction or querying of the *What system*. In order to produce a string label, the teacher should have separate access to the environment as well as knowledge of where the model is looking; the camera/sensor position is given by the Camera movement actuator (120). The following table summarizes the input/output relationship of the *Teacher*.

[0226] **Input:**

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation

NVIDIA Ex1002 Page 348 of 1081

Sensor (Eye) Position	Vector	1 (Cartesian, Ego-centered)	Unsigned, Byte	2		110 (*O*)

[0227] Output:

Name	Data Type	Data Dimensions	Element Type	Element Dimensions	Variable	Equation
External Label	String	1				N/A

[0228] Alternative OpenEye *What* System: spatio-temporal classification

[0229] An alternative classifier architecture for (280) and (290) is presented here which includes classifying information at different spatial and temporal scales. Low-level changes in input (e.g., image) features are often associated with a stable higher-order category. For instance, in vision, object rotation around its axis, or scale distortion by a moving observer, causes changes in low-level features while a top-level category (e.g., the representation of the object identity) remains stable. Hierarchical models have the advantage of capturing these relationships and autonomously building categorical/causal relationships among low and high-level features. The size of the hierarchy scales in response to the changing complexity of the underlying sense data.

[0230] OpenEye classifiers in the *What System* can be expanded to include temporal relationships between sensory or higher-order patterns. The proposed method provides a hierarchical biologically-inspired classification and prediction algorithm system for spatio-temporal classification that further extends the Adaptive Resonance Theory to enable categorization and prediction of temporal sequences in real time through the following innovations:

[0231] A *predictive subsystem*, activated upon recognition of a currently presented category, which learns to signal and prime for the most common input that appears next in a sequence. This predictive subsystem robustly learns short sequences by operating on categories of events rather than individual input patterns, and the learned knowledge is easily extracted as a transition probability matrix among learned clusters of sensor states.

[0232] A new *learning rule* for the predictive subsystem that allows for temporally delayed learning. This learning rule will couple the temporal delay of predictive layer

NVIDIA Ex1002 Page 349 of 1081

learning to an ART parameter, vigilance, which controls the granularity of learned categories. This same component will also control the rate of learning relative to the input data stream, providing a way of adaptively partitioning a temporally continuous input into discrete sequences of events. The combination of a new predictive subsystem and a new learning rule will lead to a novel ART implementation, named temporal ART, or tART.

[0233] tART modules can be stacked into a *hierarchy* able to simultaneously learn at multiple scales of complexity in time and space.

[0234] FIGS. 5A and 5B illustrate an implementation of the tART model. FIG. 5A shows that the tART model expands on ART systems by adding a layer of nodes (F3) paired by direct input with category nodes in layer F2. Connectors with arrowheads are direct excitatory connections; connectors with triangles are adaptive weights that store learned patterns. (i) Upon presentation of an input pattern that activates F1 nodes to various degrees (fill color of circles in F1 rectangular box, darker is more active), category nodes in F2 compete in the ART search cycle to find the best match. (ii) Once a single category node is active after F2 competition, its corresponding sequence learning node in F3 is activated. (iii) Upon presentation of the next pattern, the feedback connection weights from layer F3 to F2 can prime F2 during the ART search cycle to suggest a preferred category in ambiguous cases. (iv) When the second input pattern is categorized, a connection between the previously active F3 node and the currently active F2 node is strengthened to learn a temporal sequence of categories.

[0235] FIG. 5B shows a hierarchy of tART modules that can learn increasingly complex patterns and increasingly long category sequences. The categories at higher levels learn a compressed sequence structure that is abstracted from the low-level patterns. The bold arrows show how a clear sequence presented at module 1 can activate the sense making module 2 at the top of the hierarchy, which can then prime an ambiguous pattern presented at module 3. For example, if module 1 is presented clear video images of a ball moving through the viewing field, then it may activate a context in module 2 (e.g. a particular sport being played) that helps understand ambiguous video images such an ball moving behind an occluding object.

[0236] Translating ART into a modular algorithm that learns both to cluster inputs and to predict upcoming category representations requires several improvements over current implementations. ART models include input fields (the F1 layer), where input features are

NVIDIA Ex1002 Page 350 of 1081

represented, and a coding field (layer F2), where neurons compete that are responsible for learning compressed categories of features. A vigilance subsystem is responsible for regulating the granularity of learning, where learning of bottom-up and top-down (feedback) representations occurs only when the match between input and expectations satisfies the vigilance parameter. The match of an input to an existing category or the creation of a new category is regulated by the ART search cycle, where an insufficient match between F1 input and F2 category representation triggers a reset that silences the mismatched category and restarts the competition among F2 category nodes. This innovation extends the ART framework in the following ways:

[0237] ***Creation of a temporal prediction layer.*** Current ART implementations only learn temporal sequences if the sensor input field F1 is preprocessed to contain temporal information. A layer of model cells can be added, F3, where each cell is paired with and activated by a single F2 category node at the end of an ART category search cycle (FIG. 5A). Each F3 node has adaptive feedback connections back to all F2 nodes that learns to predict the winning category of the next presented input. These same F3→F2 connections represent learned sequencing knowledge that can be read by an end user as expected future sensor inputs. A learning rule gates learning by the finished categorization of two input patterns presented at different times.

[0238] ***Control effect of temporal prediction on categorization.*** Once a pair of input categories is stored as a sequence in the F3→F2 connection weights, the expectation created by the presence of the first input can be used to resolve, or disambiguate, the categorization of a noisily presented second input. This specific effect is designed for the F3 layer to prime an expected F2 category in a way that allows the expected category to be chosen in ambiguous cases. The priming effect scales with the vigilance parameter, which controls the strictness of the category matching criterion across inputs. A high vigilance value requires the input to match a category only if the inputs are similar to the category representation (stored in the F2→F1 connectivity) *and* if the category is expected by the predictive layer (stored in the F3→F2 connectivity). This allows category size and specificity to adaptively vary both across features in the input sensor and across time.

[0239] Category outputs from low-level ART units can easily be used as the input to high-level units, but there is no current way for a high-level category to prime low-level ART implementations via feedback connections. The priming from high-level units to low-level ones establishes a slow context which can help categorize a noisy signal presented on a fast

NVIDIA Ex1002 Page 351 of 1081

timescale. This same priming can also transfer category information from an unambiguous low-level pattern through the high-level unit back to a different low-level unit presented with an ambiguous pattern (FIG. 5b).

[0240] The addition of the F3 layer in the tART model allows high-level ART units in a hierarchy to cluster low-level category *sequences*, rather than simply clustering category *patterns* across low-level units. Where a low-level F3 layer can only learn pairs of categories, a high-level F3 unit can learn a longer sequence. A sequence can be translated into spatial pattern through a temporal decay that produces a gradient where the most recent item is most active and earlier items are increasingly less active. The rate of gradient decay can be regulated to maximize high-level learning rates while minimizing predictive priming interference at low levels.

[0241] The knowledge extracted from tART can provide information about the future trajectory/state of sensory stimuli. For example, the F2→F1 weight matrix of module 1 in FIG. 5B can be displayed as a set of learned input category clusters, e.g. a set of ball trajectories on a playing field. The F3→F2 weight matrix of module 1 can be displayed as a set of predictions: if the ball begins a trajectory, it can be expected to be followed by certain other paths. The F2→F1 weight matrix in module 2 of FIG. 5b codes for sets of trajectory sequences that create a context, such as a particular team playing an aggressive or defensive strategy. The adaptive F2→F2 feedback matrix from module 2 to module 1 can be read as how context changes the set of expected ball trajectories. The comparison, for example, of the F2→F1 matrix of module 1 and the multiplicatively combined (F2→F2)x(F2→F1) matrix shows how different contexts (e.g. playing strategies) produce fine-tuned corrections to the model's expected input patterns (ball trajectories).

[0242] Multimodal processing in OpenSense: focus of attention, amplification of pertinent features, fusion

[0243] FIG. 6 exemplifies multimodal processing in a practical 3-sensor case. FIG. 6A shows that Stage 3 of OpenSense includes a high-level *Where* pathway which combines information from the *Where* pathways from OpenEye, OpenEar, and OpenRadio. Sensory-specific *Where* systems bid for attentional focus. In this example, *Where* modules have multiple targets, each to be visited sequentially. The high-level *Where* system determines which target should be visited first via a competition/choice mechanism, and additionally biases the focus of attention of OpenSense to facilitate binding of coherent physical signals,

NVIDIA Ex1002 Page 352 of 1081

as shown in FIG. 6B. Feedback from the high-level *Where* system enhances physical signals from coherent objects, and suppresses the ones from incoherent ones. This in turn allows the *What* system to learn sensory signals belonging to the same physical object.

[0244] The example in FIGS. 6A and 6B shows how the high-level *Where* system in OpenSense determines which target should be visited first via a competition/choice mechanism, and how it additionally biases the focus of attention of OpenSense to facilitate binding of coherent physical signals. Feedback from the high-level *Where* system (700) enhances physical signals from coherent objects, and suppresses the ones from incoherent ones. This in turn allows the *What* system (800) to learn sensory signals belonging to the same physical object.

This stage allows a high-level *What* system to fuse coherent multi-sensory information, namely sensor signal pertaining to the same physical object. This process allows creating unique objects categories that map spatially-defined visual, auditory, and radio signals to a unique object representation.

[0245] FIG. 7 shows anomaly detection based on raw data match/mismatch. The matching process originates when the multimodal node corresponding to the current focus of attention (e.g., "Jeff") activates, via feedback, raw sensory expected representation in OpenEye, OpenEar, and OpenRadio to match actual raw data. The feedback hemicycle allows identification of anomalies in the associative category. In the example, Jeff's video and audio signatures are within tolerance of the prototype, but his radio signal is significantly different from what expected. The mismatch can be picked up to generate anomaly alerts. Anomaly alerts can be used by analysts to focus attention on changes in the scene or objects in the scene. The benefit of these alerts can be to reduce the amount of data an analyst needs to look at to find out what has changed in the environment.

[0246] The high-level *What* system also projects back via feedback connections each sensor input stage to match object expectation with low-level, raw sensor data and generate anomaly alerts (FIG. 7).

[0247] FIG. 8 shows example of biased data collection. The auditory identification of Jeff biases the visual system to collect more information about Jeff with other modalities. The localization of Jeff by the auditory system can provide spatial cues on where to orient other sensors in space.

NVIDIA Ex1002 Page 353 of 1081

[0248] OpenSense, and its visual instantiation OpenEye, capture all these factures in a single framework.

[0249] FIG. 9 shows how the *What* system takes advantage of the *Where* system processing to fuse (only) pertinent (namely, coherent, or coming from the same physical object) information into a single semantically-labeled category. More specifically, FIG. 9 shows a search driven by the presence of a search target—in this, case, looking for a specific person in a scene (e.g. Nancy). The low-level features linked with a specific learned object in each modality (vision and audition are shown here, but the same reasoning applies to other modalities) are amplified in the input stream. This facilitates search of the specific object of interest, as the saliency of the object focus of attention can be amplified thanks to the up-regulation (boosting) of the input. A mechanism to enable this boosting can comprise an additive combination between input stream and features learned by the *What* and *Where* system.

[0250] OpenSense can be also expanded to include disambiguation between sensory scenes, as an extension of disambiguation between competing object identity (Sherbakov et al., 2013a, b). OpenEye next saccade location is driven, among other factors (e.g., explicit search target, or bottom-up saliency) by the *What* system in order to disambiguate uncertainly between sensory input and internal expectation of an object. Similarly to the within-object disambiguation strategies described in Sherbakov et al. (2013a, b), a given object view can be linked to scene maps, where OpenEye or OpenSense build image-centric or world-centric (allocentric) maps of the visual environment by placing object in an map and associating that map with a specific name (e.g., garage, or bathroom). At each camera movement, As per the within-object disambiguation strategies described in Sherbakov et al. (2013a,b), each camera movement in OpenEye simultaneously activates the memory of learned scenes in which the object has been known to exist. E.g., the view of a cup simultaneously activates scene identities "office", "living room", and "kitchen". Each scene is characterized by a scene map where OpenEye has formed size invariant maps of the objects which have been seen in each scene. These objects are compared in the spatial neighborhood of the current object in the center of the camera field, and the object or objects that differ the most among the different maps of each activated scene at that location are selected as the target for a search to disambiguate the scene. This engages the explicit object search described in FIG. 9. This will in turn help to disambiguate the scene further, until the different exceeds a threshold, which can be set by a neural field (e.g., recurrent competitive field).

NVIDIA Ex1002 Page 354 of 1081

[0251] Integration with virtual environment and robotic platforms

[0252] OpenEye has been reduced to practice by integration with a Virtual Environment (VE) and a robotic platform. The VE that provides sensory input to the *What* and *Where* systems and allows to execution of motor commands (pan/tilt simulated camera).

[0253] Implementation in a virtual environment – Virt-U

[0254] OpenEye can interface with the Virtual Environment framework Virt-U (Figure 12). Virt-U (1150) creates a framework that connects Neural Modeling Environment (1100) and virtual character acting in a 3-D virtual environment controlled by a physical game engine (1200). This connection enables running complex neural modeling simulations which involve multiple sensory modalities and diverse motor control. The latter, in turn, enables to simulate behavior that lead to acquisition of sensory data, which are not determined but, in fact are the result of interaction between the brain and the environment where brain leaves.

[0255] Virt-U Architectural Principles

[0256] Virt-U architectural principles abstract the interface that connects a virtual world with its neural modeling environment. In order to segregate dependencies of the neural modeling environment from those of the virtual environment engine, the Virt-U architecture was designed to include two major layers: the ***virtual environment layer*** (1170) and the ***proxy layer*** (1170). The virtual environment (VE) layer abstracts the virtual environment engine, while the proxy layer delivers sensory information to the brain and extracts neural information from the brain in order to perform the behavior. An important benefit of this approach is that it supports functioning of Virt-U in two distinct modes:

[0257] ***Simulation*** – when the behavior of the animat is controlled by the brain model (FIG. 11), and

[0258] ***No-brainer*** – no brain is attached (FIG. 12).

[0259] Virt-U considers the *physical* world to be populated by ***virtual objects***. These virtual objects can be animated (brained) and non-animated (brainless). Brained objects, called animats, can be controlled by a neural model formulated using a native neural modeling language, and then executed by a neural modeling engine. An animat's brains receives sensory (e.g., visual) and proprioceptive signals and can then exhibit a behavior that follows the laws of physics to a predefined level. Animats can be either virtual animals (e.g., a rat) or vehicles (e.g., a rover), depending on how their motor functions are controlled, but a

NVIDIA Ex1002 Page 355 of 1081

brain in both cases controls navigation. Brainless objects, called items, are entirely controlled by the virtual environment and obey its physics. Items include rocks, buildings, trees, etc. and can be considered a source for all kinds of sensory.

[0260] Virt-U considers an animat to be a collection of ***sensory organs*** and ***animat controls***. Sensory organs may in turn include sub-sensory organs, which ultimately can be connected with ***virtual sensors*** dedicated to collecting sensory information. For example, an animat may have a sensory organ called "eyes" that includes individual eyes, where each eye contains facets connected with virtual cameras. All sensory organs are responsible for maintaining and updating sensory information for the sub-organs they contain thus allowing for recursive sensory data update to be performed on an animat. Animat controls constitute outgoing interfaces that are accessed by a brain to perform the required behavior. Sensory organs and animat controls expose specific incoming and outgoing interfaces to be accessed from the Virt-U proxy layer in order to supply and deliver sensory and motor control information.

[0261] On the neural modeling side, ***proxy organ controllers*** perform access to the VE. These controllers retrieve and supply neural data by accessing a specific organ controllers' incoming and outgoing interfaces. Thus proxy controllers abstract internal knowledge of sensory organs from the neural model by acting as hardware drivers within the neural modeling environment. In simulation mode, Virt-U creates the virtual world based on an XML description. This description is sufficient to define the world, the animats with all their sensory organs and controls, and all virtual objects with their associated rewards. This XML-based virtual world description ensures Virt-U portability between various neural modeling environments.

[0262] FIG. 12 shows Virt-U's how complex sensory data are collected and updated. In particular, FIG. 12 pertains to modifications to the *Where* system diagram of FIG. 2B in order to exploit Virt-U. One modification includes a rapid segregation of foreground and background based on texture and other information. This modification can be implemented as a rapid segregation module that receives an input from the segmentation module 180 and provides an output to the figure/ground segmentation module 210 shown in FIG. 2B.

[0263] **Implementation in a robotic platform**

NVIDIA Ex1002 Page 356 of 1081

[0264] OpenEye has been tested in a physical robotic platform. The equipment developed for this assessment includes an external computer running OpenEye, a physical robot, and its sensors and actuators.

[0265] The robotic platform includes 2 Lego Mindstorms NXTs outfitted with 7 sensors and 4 motors. In this implementation, the NXTs communicate with a single-board computer development platform (e.g., Pandaboard), which controls communications with a OpenEye client machine. Other robotic platforms include, but are not limited to, the following robots: Romotive Romo, Parrot AR Drone, iRobot Create, Vex. The Pandaboard runs an asynchronous server that listens for commands from the OpenEye client while reading data from the NXTs and camera. When data is received from an NXT, it is repackaged and sent over UDP to the OpenEye client. The Pandaboard utilizes OpenCV to process image data from a USB camera and preprocesses/broadcasts it to Cog over TCP through an onboard router connected to the board through a Cat5e cable. The OpenEye client is capable of sending movement commands and can also schedule tasks on the NXTs. The client listens for data from each of the NXTs. FIG. 13 shows a hardware diagram and the protocols used to connect the devices, and FIG. 14 illustrates the main objects within the software of the system and their respective communication streams.

[0266] **Introduction of a Coarse Observer for OpenSense and OpenEye**

[0267] OpenSense, as well as individual sensory instantiation OpenEye, implement a biologically-inspired approach for sensory scene processing. These systems may implement detailed object recognition, or a *fine observer* system. The additional methods described herein are to complement the fine observer with a *coarse observer* that may quickly assess the scene and direct the fine observer or image processing system to the most salient or interesting regions or objects of the scene. The implementation below is initially described in the context of, but should not be limited to, visual processing on an unmanned ground vehicle (UGV). With the addition of the biologically-inspired coarse observer, at least the following capabilities may be added (see FIG. 15):

[0268] ***Fast, parallel pre-attentive segmentation*** of the visual scene into *scene macro-areas* (e.g. sky, grass, roads) to bias the system's *active focus of attention*, described below, to the most informative regions or objects in the UGV field of view (e.g. people, vehicles). Scene macro-areas may be *learned and semantically labeled*, allowing scalability of the

NVIDIA Ex1002 Page 357 of 1081

system to previously unseen environments. Scene macro-areas may not be discarded but rather may have lower priority for more in depth visual processing.

[0269] ***Active focus of attention*** through serial deployment of limited computational resources in order to sample high-priority areas and objects in a visual scene that are identified by the coarse observer. This sampling results in rapid and accurate labeling and identification of objects, events, and situations that require fast adaptation of goals, priorities, and plans. This labeling and identification is performed by a *fine observer*, already under development.

[0270] ***Situational awareness*** for high-level reasoning in the form of incremental indexing of the UGV sensory stream as the vehicle transverse the environment. Each frame may be associated with image metadata comprising identity of scene macro-areas, object of interests, and their position. The system may *extract this knowledge* for high-level UGV reasoning.

[0271] FIG. 15 shows how a coarse observer initially segments large scene areas (e.g., sky and ground macro-areas), and bias a fine observer to *first* scan other areas not classified as macro-areas, or unknown macro-areas, which can potentially contain useful objects (e.g., a road sign, a person, or a UAV).

[0272] FIG. 16 describes the coarse observer and its interactions with the fine observer. The coarse observer learns to produce a rough and fast semantic labeling of a viewed scene. The system will rapidly segment large images into texture regions by summarizing small image regions into a standard descriptive format. These small regions will then be glued into *scene macro-areas* that are given semantic labels such as sky, ground, and horizon. These texture regions and their labels will then be used to direct the limited processing power of the fine observer for extracting object identity, and they will be summarized in scene metadata information.

[0273] FIG. 16 shows coarse and fine observer interactions. Sensor data acquired from the UGV (1) is preprocessed (2). The fast scene preprocessing in the coarse observer (black rectangles modules in dashed area) quickly classifies scene macro-areas (3) to focus attention (4) on the object recognition system (5) in the fine observer (red rounded rectangles modules in dotted area) and possibly gather more information to classify objects of interest and to influence sensor control (6). Recognized scene macro-areas and objects are combined in each frame into scene metadata (7), which forms the basis of knowledge for UGV higher-level

NVIDIA Ex1002 Page 358 of 1081

reasoning. A teacher (8) provides labels to new scene macro-areas and objects to scale the system to new environments. The fine observer leverages OpenSense's vision models.

[0274] The technology described herein is a coarse observer that can quickly analyze high-resolution image and video data in order to produce meaningful segmentation that guides the serial deployment of more powerful but limited image analysis algorithms and the collection of scene metadata for external use. This coarse observer offers significant computational advantages by processing only interesting or relevant objects or regions.

[0275] FIG. 17 shows advantages of the coarse observer model in scene understanding. In other words, FIG. 17 shows the effect of using rapid segregation of foreground and background based on texture and other information, as shown in FIG. 12, with the *Where* system shown in FIG. 2B. *Left column:* without the coarse observer, conventional bottom-up image statistics (e.g., edge information) would bias a fine observer with limited computational resources to repeatedly sample non-informative image segments (dots connected by arrows), for instance repeatedly sampling the ground early on. *Center column:* a coarse observer module performs a fast analysis of image statistics, groups image segments into known scene macro-areas (sky, ground), and biases the fine observer to sample more informative image segments first, for instance sampling initially the soldiers, and then the ground. *Right column:* the coarse observer can be trained to recognize several macro-areas, allowing scalability of the system to different environments.

[0276] ***Texture extraction for scene macro-areas module***

[0277] Meaningful image segmentation based on learned labels can help to intelligently optimize limited visual processing power. OpenSense may facilitate fast image segmentation into *scene macro-areas* that can be toned by learned semantic labels. This segmentation method may collect image statistics over many small regions and group adjacent regions with similar statistics into scene macro-areas. More specifically, the segmentation involves:

[0278] *Carving up the scene into a grid of equally spaced boxes* (FIG. 17, yellow boxes) *and extracting a stereotyped feature vector from each sub-area*. Oriented FAST and Rotated Brief descriptors (ORB; Rublee et al., 2011) are an example of a binary feature descriptor of the region around an image point that remains unchanged by changes in lighting, orientation, and position, but the points are chosen only at the most "point-like" pixels in the image. This descriptor will be adapted to describe at least one point within each small region in the image grid. The resulting feature vectors are collected separately for each area.

NVIDIA Ex1002 Page 359 of 1081

[0279] *Grouping together regions of similar statistics into scene macro-areas.* Perceptual filling-in (Pessoa et al., 1998), often implemented by simulating the diffusion of a material with speed proportional to region feature similarity, provides a simple method for constructing regions with approximately homogeneous features.

[0280] Assigning a single feature vector to each scene macro-area, the average of feature vectors of its constituent image regions, to be used to categorize and label the macro-area. This macro-area feature vector can also be compared to each subregion's feature vector in order to find image anomalies (typicality measure).

[0281] ***Object and situation-based control of attention***

[0282] An attentionally-guided interface between scene macro-areas and the fine observer will allow for the simultaneous fulfillment of multiple goals. For example, if the fine observer requests more views of an ambiguous object, this must be balanced against the need to focus on a particular region as signaled by the scene macro-areas module. This interface will be a focus of attention (FIG. 16, box 4) or field over the image that signals viewing priority according to scene macro-area. This map can then be combined with the fine observer's internal measure of viewing priority in a way that considers multiple viewing goals.

[0283] ***Knowledge extraction module***

[0284] The user-readable result of this work will be a high-level compressed representation of the scene as viewed by the artificial visual system—extracted knowledge in the form of scene metadata (FIG. 16, box 7). This will allow the system to maintain *situational awareness*. The coarse observer will efficiently direct the fine observer towards objects of interest, whose identity and position will be recorded along with the identity and extent of scene macro-areas. Created scene metadata can then be used by an operator or high-level context processor that directs situational awareness.

[0285] FIG. 18A illustrates an example of a typical run of the UGV in a rural environment, and FIG. 18B shows examples of scene metadata provided by the system on each frame. In FIG. 18A, the UGV drives through a set path between start and goal location in the presence of static/moving objects. And FIG. 18B shows that at each frame, the system provides scene metadata for both large areas and objects, along with their location.

[0286] To fully comply with real-time vision in a mobile platform, the UGV must be able to detect the temporal continuity of objects classified by the fine observer incorporating robot

NVIDIA Ex1002 Page 360 of 1081

or object motion. A non-limiting example of temporal continuity is exemplified in FIG. 19. This implementation for objects, combined with the temporal continuity of scene macro-areas may prevent unnecessary duplication of object records in scene metadata.

[0287] FIG. 19 illustrates temporal continuity in the fine observer. Using temporal continuity in the *Where* system can prevent the problem of having to reclassify the same object multiple time as the robot moves around. The identity of classified objects (e.g., rocks *a*, *b* classified in a simulated planetary environment) and their position is predicted at each of the 3 frames to anticipate where to look next and build a coherent visual world as the robot moves in the environment. The *top-left* inset shows the fine observer model controlling the adjustment of the sample location at each frame taking into account motion of the robot, its sensor, and perceived objects. The *bottom-right* inset shows the temporal continuity of the two space-variant log-polar representation of one of the rocks (red outline) in two successive frames where rock identity is consistently the same across the two frames.

[0288] ***Integration with fine observer***

[0289] The results from both the fine and coarse observers will enhance the efficiency and performance of each other, which is optimized by close integration of these two systems. Once a candidate set of sensors is chosen, the input from these sensors will be formatted for efficient learning and recognition by the fine observer. The method of directing the fine observer's focus of attention will be enhanced by creating a mixing rule between pre-existing bottom-up saliency for object learning and the top-down saliency field from the coarse observer.

[0290] The coarse observer should be able to learn from image examples in order to build internal representations of objects and image regions so that it can both efficiently process new and complicated visual contexts and communicate with an end user. The text describes how the coarse observer will be trained to recognize canonical scene macro-areas as well as give them human-readable and semantically relevant labels and viewing priority weightings.

[0291] ***Adapt object clustering system to image macro-areas***

[0292] Unsupervised learning models will be applied to the coarse observer in order to cluster and categorize observed scene macro-area feature vectors. This clustering will allow commonly viewed areas, such as sky and ground, to have a persistent representation and systemic influence on the fine observer across applications and missions. Changes in the macro-area (e.g., an approaching helicopter) will not be ignored, but instead will be

NVIDIA Ex1002 Page 361 of 1081

incorporated into directing the focus of attention. In the presence of an external teacher, supervised learning will also be applied to macro-area cluster centers in order to associate regions with user-accessible semantic labels (e.g. sky, ground) and viewing priority levels. This application will reuse a combined unsupervised/supervised learning system, already developed and used in OpenSense's fine observer, in order to cluster and label scene macro-areas.

[0293] ***Train integrated system on standard image databases***

[0294] After developing a clustering and labeling system for the coarse observer, a foundation of scene macro-area clusters will be created by training the coarse observer on standard computer vision databases. This foundation will be built with the LabelMe database (Russell et al., 2008). In order to evaluate the recognition ability of the integrated system with and without the coarse observer, the fine observer will also be trained on databases of specific objects such as the KITTI dataset (Geiger, et al., 2012).

[0295] ***Bias attention towards macro-areas that are either new or interesting***

[0296] During and after learning the identity of scene macro-areas, the coarse observer will also learn to associate regions such as sky and ground with measures of viewing priority for the fine observer. This value-based association completes the behavioral circuit allowing a fast, semantically-based image segmentation to control viewing efficiency. These associations include:

[0297] A measure of *familiarity* with macro-areas that inhibits the fine observer more strongly as an area is viewed more often and frequently;

[0298] A measure of *typicality* that activates the fine observer as a small image region deviates more from its macro-area average and from the appearance learned to be normal over time;

[0299] A learned or user-defined measure of *task-based relevance* that modulates viewing priority based on a high-level situational context. For example, the presence of another vehicle as identified by the fine observer can allow the coarse observer to increase the viewing priority of macro-areas labeled as ground and road rather than sky or building.

[0300] **Conclusion**

[0301] While various inventive embodiments have been described and illustrated herein, those of ordinary skill in the art will readily envision a variety of other means and/or

NVIDIA Ex1002 Page 362 of 1081

structures for performing the function and/or obtaining the results and/or one or more of the advantages described herein, and each of such variations and/or modifications is deemed to be within the scope of the inventive embodiments described herein. More generally, those skilled in the art will readily appreciate that all parameters, dimensions, materials, and configurations described herein are meant to be exemplary and that the actual parameters, dimensions, materials, and/or configurations will depend upon the specific application or applications for which the inventive teachings is/are used. Those skilled in the art will recognize, or be able to ascertain using no more than routine experimentation, many equivalents to the specific inventive embodiments described herein. It is, therefore, to be understood that the foregoing embodiments are presented by way of example only and that, within the scope of the appended claims and equivalents thereto, inventive embodiments may be practiced otherwise than as specifically described and claimed. Inventive embodiments of the present disclosure are directed to each individual feature, system, article, material, kit, and/or method described herein. In addition, any combination of two or more such features, systems, articles, materials, kits, and/or methods, if such features, systems, articles, materials, kits, and/or methods are not mutually inconsistent, is included within the inventive scope of the present disclosure.

[0302] The above-described embodiments can be implemented in any of numerous ways. For example, embodiments of designing and making the technology disclosed herein may be implemented using hardware, software or a combination thereof. When implemented in software, the software code can be executed on any suitable processor or collection of processors, whether provided in a single computer or distributed among multiple computers.

[0303] Further, it should be appreciated that a computer may be embodied in any of a number of forms, such as a rack-mounted computer, a desktop computer, a laptop computer, or a tablet computer. Additionally, a computer may be embedded in a device not generally regarded as a computer but with suitable processing capabilities, including a Personal Digital Assistant (PDA), a smart phone or any other suitable portable or fixed electronic device.

[0304] Also, a computer may have one or more input and output devices. These devices can be used, among other things, to present a user interface. Examples of output devices that can be used to provide a user interface include printers or display screens for visual presentation of output and speakers or other sound generating devices for audible presentation of output. Examples of input devices that can be used for a user interface include keyboards, and pointing devices, such as mice, touch pads, and digitizing tablets. As

NVIDIA Ex1002 Page 363 of 1081

another example, a computer may receive input information through speech recognition or in other audible format.

[0305] Such computers may be interconnected by one or more networks in any suitable form, including a local area network or a wide area network, such as an enterprise network, and intelligent network (IN) or the Internet. Such networks may be based on any suitable technology and may operate according to any suitable protocol and may include wireless networks, wired networks or fiber optic networks.

[0306] The various methods or processes (e.g., of designing and making the coupling structures and diffractive optical elements disclosed above) outlined herein may be coded as software that is executable on one or more processors that employ any one of a variety of operating systems or platforms. Additionally, such software may be written using any of a number of suitable programming languages and/or programming or scripting tools, and also may be compiled as executable machine language code or intermediate code that is executed on a framework or virtual machine.

[0307] In this respect, various inventive concepts may be embodied as a computer readable storage medium (or multiple computer readable storage media) (e.g., a computer memory, one or more floppy discs, compact discs, optical discs, magnetic tapes, flash memories, circuit configurations in Field Programmable Gate Arrays or other semiconductor devices, or other non-transitory medium or tangible computer storage medium) encoded with one or more programs that, when executed on one or more computers or other processors, perform methods that implement the various embodiments of the invention discussed above. The computer readable medium or media can be transportable, such that the program or programs stored thereon can be loaded onto one or more different computers or other processors to implement various aspects of the present invention as discussed above.

[0308] The terms "program" or "software" are used herein in a generic sense to refer to any type of computer code or set of computer-executable instructions that can be employed to program a computer or other processor to implement various aspects of embodiments as discussed above. Additionally, it should be appreciated that according to one aspect, one or more computer programs that when executed perform methods of the present invention need not reside on a single computer or processor, but may be distributed in a modular fashion amongst a number of different computers or processors to implement various aspects of the present invention.

NVIDIA Ex1002 Page 364 of 1081

[0309] Computer-executable instructions may be in many forms, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically the functionality of the program modules may be combined or distributed as desired in various embodiments.

[0310] Also, data structures may be stored in computer-readable media in any suitable form. For simplicity of illustration, data structures may be shown to have fields that are related through location in the data structure. Such relationships may likewise be achieved by assigning storage for the fields with locations in a computer-readable medium that convey relationship between the fields. However, any suitable mechanism may be used to establish a relationship between information in fields of a data structure, including through the use of pointers, tags or other mechanisms that establish relationship between data elements.

[0311] Also, various inventive concepts may be embodied as one or more methods, of which an example has been provided. The acts performed as part of the method may be ordered in any suitable way. Accordingly, embodiments may be constructed in which acts are performed in an order different than illustrated, which may include performing some acts simultaneously, even though shown as sequential acts in illustrative embodiments.

[0312] All definitions, as defined and used herein, should be understood to control over dictionary definitions, definitions in documents incorporated by reference, and/or ordinary meanings of the defined terms.

[0313] The indefinite articles "a" and "an," as used herein in the specification and in the claims, unless clearly indicated to the contrary, should be understood to mean "at least one."

[0314] The phrase "and/or," as used herein in the specification and in the claims, should be understood to mean "either or both" of the elements so conjoined, i.e., elements that are conjunctively present in some cases and disjunctively present in other cases. Multiple elements listed with "and/or" should be construed in the same fashion, i.e., "one or more" of the elements so conjoined. Other elements may optionally be present other than the elements specifically identified by the "and/or" clause, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, a reference to "A and/or B", when used in conjunction with open-ended language such as "comprising" can refer, in one embodiment, to A only (optionally including elements other than B); in another embodiment,

NVIDIA Ex1002 Page 365 of 1081

to B only (optionally including elements other than A); in yet another embodiment, to both A and B (optionally including other elements); etc.

[0315] As used herein in the specification and in the claims, "or" should be understood to have the same meaning as "and/or" as defined above. For example, when separating items in a list, "or" or "and/or" shall be interpreted as being inclusive, i.e., the inclusion of at least one, but also including more than one, of a number or list of elements, and, optionally, additional unlisted items. Only terms clearly indicated to the contrary, such as "only one of" or "exactly one of," or, when used in the claims, "consisting of," will refer to the inclusion of exactly one element of a number or list of elements. In general, the term "or" as used herein shall only be interpreted as indicating exclusive alternatives (i.e. "one or the other but not both") when preceded by terms of exclusivity, such as "either," "one of," "only one of," or "exactly one of." "Consisting essentially of," when used in the claims, shall have its ordinary meaning as used in the field of patent law.

[0316] As used herein in the specification and in the claims, the phrase "at least one," in reference to a list of one or more elements, should be understood to mean at least one element selected from any one or more of the elements in the list of elements, but not necessarily including at least one of each and every element specifically listed within the list of elements and not excluding any combinations of elements in the list of elements. This definition also allows that elements may optionally be present other than the elements specifically identified within the list of elements to which the phrase "at least one" refers, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, "at least one of A and B" (or, equivalently, "at least one of A or B," or, equivalently "at least one of A and/or B") can refer, in one embodiment, to at least one, optionally including more than one, A, with no B present (and optionally including elements other than B); in another embodiment, to at least one, optionally including more than one, B, with no A present (and optionally including elements other than A); in yet another embodiment, to at least one, optionally including more than one, A, and at least one, optionally including more than one, B (and optionally including other elements); etc.

[0317] In the claims, as well as in the specification above, all transitional phrases such as "comprising," "including," "carrying," "having," "containing," "involving," "holding," "composed of," and the like are to be understood to be open-ended, i.e., to mean including but not limited to. Only the transitional phrases "consisting of" and "consisting essentially

NVIDIA Ex1002 Page 366 of 1081

of" shall be closed or semi-closed transitional phrases, respectively, as set forth in the United States Patent Office Manual of Patent Examining Procedures, Section 2111.03.

[0318] References

[0319] The following references are incorporated herein by reference in their respecitve entireties:

Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., & Ogden, J. M. (1984). Pyramid methods in image processing. RCA engineer, 29(6), 33-41.

Baraldi, A. and Alpaydin, E. (1998). Simplified ART: A new class of ART algorithms. International Computer Science Institute, Berkeley, CA, TR-98-004, 1998.

Baraldi, A. and Alpaydin, E. (2002). Constructive feedforward ART clustering networks — Part I. *IEEE Transactions on Neural Networks* 13(3), 645–661.

Baraldi, A. and Parmiggiani, F. (1997). Fuzzy combination of Kohonen's and ART neural network models to detect statistical regularities in a random sequence of multi-valued input patterns. In *International Conference on Neural Networks*, IEEE.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives.

Besl, P. J., & Jain, R. C. (1985). Three-dimensional object recognition. ACM Computing Surveys (CSUR), 17(1), 75-145.

Bradski, G., & Grossberg, S. (1995). Fast-learning VIEWNET architectures for recognizing three-dimensional objects from multiple two-dimensional views. *Neural Networks, 8* (7-8), 1053-1080.

Canny, J., *A (1986)* Computational Approach To Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698.Carpenter, G.A. and Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing* 37, 54–115.

Carpenter, G.A., & Grossberg, S. (1987). A massively parallel architecture for a self-organizing

Carpenter, G.A., and Grossberg, S. (1995). Adaptive resonance theory (ART). In M. Arbib (Ed.), The handbook of brain theory and neural networks. (pp. 79–82). Cambridge, M.A.: MIT press.

NVIDIA Ex1002 Page 367 of 1081

Carpenter, G.A., Grossberg, S. and Rosen, D.B. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks* 4, 759–771.

Dosher, B.A., and Lu, Z.L. (2010). Mechanisms of perceptual attention in precuing of location. Vision Res., 40(10–12). 1269–1292.

Fazl, A., Grossberg, S., and Mingolla, E. (2009). View-invariant object category learning, recognition, and search: How spatial and object attention are coordinated using surface-based attentional shrouds. *Cognitive Psychology* 58, 1–48.

Földiák, P. (1990). Forming sparse representations by local anti-Hebbian learning, Biological Cybernetics, vol. 64, pp. 165-170.

Friston K., Adams R., Perrinet L., & Breakspear M. (2012). Perceptions as hypotheses: saccades as experiments. Frontiers in Psychology, 3 (151), 1-20.

Geiger A, Lenz P, and Urtasun R (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR)*, Providence, USA.

George, D. and Hawkins, J. (2009). Towards a mathematical theory of cortical micro-circuits. *PLoS Computational Biology* 5(10), 1–26.

Grossberg, S. (1973). Contour enhancement, short-term memory, and constancies in reverberating neural networks. *Studies in Applied Mathematics* 52, 213–257.

Grossberg, S., and Huang, T.R. (2009). ARTSCENE: A neural system for natural scene classification. *Journal of Vision*, *9* (4), 6.1–19. doi:10.1167/9.4.6

Grossberg, S., and Versace, M. (2008) Spikes, synchrony, and attentive learning by laminar thalamocortical circuits. *Brain Research,* 1218C, 278-312 *[Authors listed alphabetically]*.

Hinton, G. E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18, 1527–1554.

Itti, L., and Koch, C. (2001). Computational modelling of visual attention. *Nature Reviews Neuroscience, 2* (3), 194-203.

Itti, L., Koch, C., and Niebur, E. (1998). A Model of Saliency-Based Visual Attention for Rapid Scene Analysis, 1–6.

NVIDIA Ex1002 Page 368 of 1081

Jarrett, K., Kavukcuoglu, K., Ranzato, M. A., & LeCun, Y. (2009, September). What is the best multi-stage architecture for object recognition?. In Computer Vision, 2009 IEEE 12th International Conference on (pp. 2146-2153). IEEE.

Kowler, E. (2011). Eye movements: The past 25years. *Vision Research*, *51*(13), 1457–1483. doi:10.1016/j.visres.2010.12.014

Larochelle H., & Hinton G. (2012). Learning to combine foveal glimpses with a third-order Boltzmann machine. NIPS 2010, 1243-1251.

LeCun, Y., Kavukcuoglu, K., & Farabet, C. (2010, May). Convolutional networks and applications in vision. In Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on (pp. 253-256). IEEE.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755):788-791.

Lee, D. D., and Seung, H. S. (1997). "Unsupervised learning by convex and conic coding." Advances in Neural Information Processing Systems, 9.

Lowe, D.G.(2004). Distinctive Image Features from Scale-Invariant Keypoints. Journal International Journal of Computer Vision archive Volume 60, 2, 91 – 110.

Lu, Z.L., Liu, J., and Dosher, B.A. (2010) Modeling mechanisms of perceptual learning with augmented Hebbian re- weighting. *Vision Research*, 50(4). 375–390.

Mishkin M, Ungerleider LG. (1982). "Contribution of striate inputs to the visuospatial functions of parieto-preoccipital cortex in monkeys," Behav Brain Res, 6 (1): 57–77.

Najemnik, J., and Geisler, W. (2009). Simple summation rule for optimal fixation selection in visual search. *Vision Research*, 49, 1286–1294.

neural pattern-recognition machine. Computer Vision Graphics and Image Processing, 37 (1), 54-115.

Oja, E. (1982). Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology* 15(3), 267–273.

Pessoa L, Thompson E, and Noë A (1998). Finding out about filling-in: A guide to perceptual completion for visual science and the philosophy of perception. *Behavioral and Brain Sciences 21*(6), 723–748.

NVIDIA Ex1002 Page 369 of 1081

Raijmakers, M.E.J., and Molenaar, P. (1997). Exact ART: A complete implementation of an ART network Neural networks 10 (4), 649-669.

Ranzato, M. A., Huang, F. J., Boureau, Y. L., & Lecun, Y. (2007, June). Unsupervised learning of invariant feature hierarchies with applications to object recognition. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on (pp. 1-8). IEEE.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience, 2* (11), 1019-1025.

Riesenhuber, M., & Poggio, T. (2000). Models of object recognition. Nature neuroscience, 3, 1199-1204.

Rublee E, Rabaud V, Konolige K, and Bradski G (2011). ORB: An efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision (ICCV)*, 2564–2571.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision (ICCV) 2011*, 2564–2571.

Rumelhart D., Hinton G., and Williams, R. (1986). Learning internal representations by error propagation. In *Parallel distributed processing: explorations in the microstructure of cognition,* vol. 1, MIT Press.

Russell B, Torralba A, Murphy K, and Freeman W T (2008). LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision 77*(1–3), 157–173.

Salakhutdinov, R., & Hinton, G. E. (2009). Deep boltzmann machines. In International Conference on Artificial Intelligence and Statistics (pp. 448-455).

Seibert, M., & Waxman, A.M. (1992). Adaptive 3-D Object Recognition from Multiple Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14* (2), 107-124.

Sherbakov, L., Livitz, G., Sohail, A., Gorchetchnikov, A., Mingolla, E., Ames, H., and Versace, M. (2013a) CogEye: An online active vision system that disambiguates and recognizes objects. NeuComp 2013.

Sherbakov, L., Livitz, G., Sohail, A., Gorchetchnikov, A., Mingolla, E., Ames, H., and Versace, M (2013b) A computational model of the role of eye-movements in object disambiguation. *Cosyne,* Feb 28 - Mar 3, 2013. Salt Lake City, UT, USA.

NVIDIA Ex1002 Page 370 of 1081

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhartand J. L. McClelland, editors, Parallel Distributed Processing, volume 1, chapter 6, pages 194–281. MIT Press, Cambridge.

Spratling, M. W. (2008). Predictive coding as a model of biased competition in visual attention. Vision Research, 48(12):1391-1408.

Spratling, M. W. (2012). Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. Neural Computation, 24(1):60-103.

Spratling, M. W., De Meyer, K., and Kompass, R. (2009). Unsupervised learning of overlapping image components using divisive input modulation. Computational intelligence and neuroscience.

Tong, F., Ze-Nian Li, (1995). Reciprocal-wedge transform for space-variant sensing," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.17, no.5, pp.500 - 51. doi: 10.1109/34.391393

Torralba, A., Oliva, A., Castelhano, M.S., Henderson, J.M. (2006). Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological Review*, 113(4).766–786.

Versace, M. (2006) From spikes to interareal synchrony: how attentive matching and resonance control learning and information processing by laminar thalamocortical circuits. NSF Science of Learning Centers PI Meeting, Washington, DC, USA.

Webster, Bachevalier, Ungerleider (1994). Connections of IT areas TEO and TE with parietal and frontal cortex in macaque monkeys. Cerebal Cortex, 4(5), 470-483.

NVIDIA Ex1002 Page 371 of 1081

CLAIMS

1. A system for automatically locating and identifying an object in an environment, the system comprising:

at least one sensor to acquire sensor data representing of at least a portion of the object;

a spatial attention module, operably coupled to the at least one sensor, to produce a foveated representation of the object based at least in part on the sensor data, to track a position of the object within the environment based at least in part on the foveated representation, and to select another portion of the environment to be sensed by the at least one sensor based at least in part on the foveated representation of the object; and

a semantics module, operably coupled to the spatial attention module, to determine an identity of the object based at least in part on the foveated representation of the object.

2. The system of claim 1, wherein the at least one sensor comprises an image sensor to acquire at least one image of the at least a portion of the environment.

3. The system of claim 2, further comprising:

at least one actuator, operably coupled to the image sensor, to provide sensor position data representative of an orientation and/or a position of the image sensor, and

wherein the spatial attention module is configured to select the other portion of the environment based at least in part on the orientation and/or a position of the image sensor.

4. The system of claim 1, wherein the spatial attention module configured to select the other portion of the environment such that the object appears at or near a center of the other portion of the environment .

5. The system of claim 1, wherein the spatial attention module comprises:

a memory to store the sensor position data representing an orientation and/or a position of the sensor; and

an inhibition of return module, operably coupled to the memory, to inhibit repeated selections of a given object based at least in part on the sensor position data stored in the memory.

6. The system of claim 1, wherein the spatial attention module comprises:

NVIDIA Ex1002 Page 372 of 1081

at least one log-polar transformation module to transform the sensor data into a log-polar representation of the environment so as to provide invariance to translation and/or rotation of the at least one sensor with respect to the object and/or so as to reduce processing time.

7. The system of claim 1, wherein the spatial attention module comprises:

a segmentation module to generate at least one contour representation of the object based at least in part on the sensor data.

8. The system of claim 7, wherein the spatial attention module further comprises:

a figure/ground segregation module, operably coupled to the segmentation module, to determine at least one spatial shroud fitting a form of the object based at least in part on the at least one contour representation of the object.

9. The system of claim 8, wherein the sensor data comprises a plurality of images and the semantics module comprises:

a view layer to group views of the object in the plurality of images based at least in part on the at least one spatial shroud ;

an object layer, operably coupled to the view layer, to map the views of the object to an object node associated with the object; and

a name layer, operably coupled to the object layer, to classify the object based at least in part on the object node.

10. The system of claim 9, wherein the view layer is operably coupled to the name layer and configured to group the views of the object based at least in part on classification of the object by the name layer.

11. The system of claim 1, wherein the semantics module is configured to learn the identity of the object based at least in part on the location of the object.

12. The system of claim 1, further comprising:

a teaching module, operably coupled to the at least one sensor and to the semantic system, to provide a label for the object, and

wherein the semantics module is configured to assign the label to the object.

NVIDIA Ex1002 Page 373 of 1081

13. A method of automatically locating and identifying an object in an environment, the method comprising:

(A) estimating a position and/or an orientation of at least one sensor with respect to the environment;

(B) acquiring, with the at least one sensor, sensor data representing at least a portion of the object;

(C) producing a foveated representation of the object based at least in part on the sensor data acquired in (B);

(D) determining an identity of the object based at least in part on the foveated representation of the object produced in (C); and

(E) selecting another portion of the environment to be sensed by the at least one sensor based at least in part on the foveated representation of the object produced in (C) and the position and/or the orientation estimated in (A);

(F) acquiring additional sensor data, with the at least one sensor, in response to selection of the other portion of the environment in (D).

14. The method of claim 13, wherein (D) comprises selecting the other portion of the environment such that the object appears at or near a center of the other portion of the environment .

15. The method of claim 13, wherein (D) comprises inhibiting repeated selections of a given portion of the environment based at least in part on the position estimated in (A).

16. The method of claim 13, wherein (D) comprises transforming the sensor data into a log-polar representation of the environment so as to provide invariance to translation and/or rotation and/or so as to reduce processing time.

17. The method of claim 13, wherein:

(A) comprises acquiring a plurality of images; and

(D) comprises generating at least one contour representation of the object based at least in part on the at least one image.

18. The method of claim 17, wherein (D) further comprises determining at least one spatial shroud fitting a form of the object based at least in part on the at least one contour representation of the object.

NVIDIA Ex1002 Page 374 of 1081

19. The method of claim 18, wherein (E) comprises:

(E1) grouping views of the object in the plurality of images based at least in part on the at least one spatial shroud ;

(E2) mapping the views of the object to an object node associated with the object; and

(E3) classifying the object based at least in part on the object node.

20. The method of claim 19, wherein (E1) comprises grouping the views of the object based at least in part on classification of the object in (E3).

21. A system for automatically locating and identifying an object in an environment, the system comprising:

a plurality of sensors to acquire multi-sensory data representative of the environment;

a plurality of first spatial attention modules, each first spatial attention module in the plurality of first spatial attention modules operably coupled to a respective sensor in the plurality of sensors and configured to produce a respective representation of the object based on a respective portion of the multi-sensory data;

a plurality of first semantics modules, each first semantics module in the plurality of first spatial attention modules operably coupled to a respective first spatial attention module in the plurality of first spatial attention modules and configured to produce respective identifications of the object based on an output from the respective first spatial attention module;

a second spatial attention module, operably coupled to the plurality of first spatial attention modules, to estimate a location of the object based at least in part on the multi-sensory representation of the environment; and

a second semantics module, operably coupled to the plurality of first semantics modules and the second spatial attention module, to identify the at least one object based in part of the respective identifications produced by the plurality of first semantics modules and the location of the object estimated by the second spatial attention module.

22. The system of claim 21, wherein the second spatial attention module is configured to dynamically allocate the plurality of sensors so as to enhance detection of the at least one object and/or to suppress noise in the multi-sensory representation of the scene.

23. The system of claim 21, wherein the second semantics module is configured to generate scene metadata representative of an identity of the at least one object and/or the at

NVIDIA Ex1002 Page 375 of 1081

least one change to the object based at least in part on the multi-sensory representation of the environment.

NVIDIA Ex1002 Page 376 of 1081

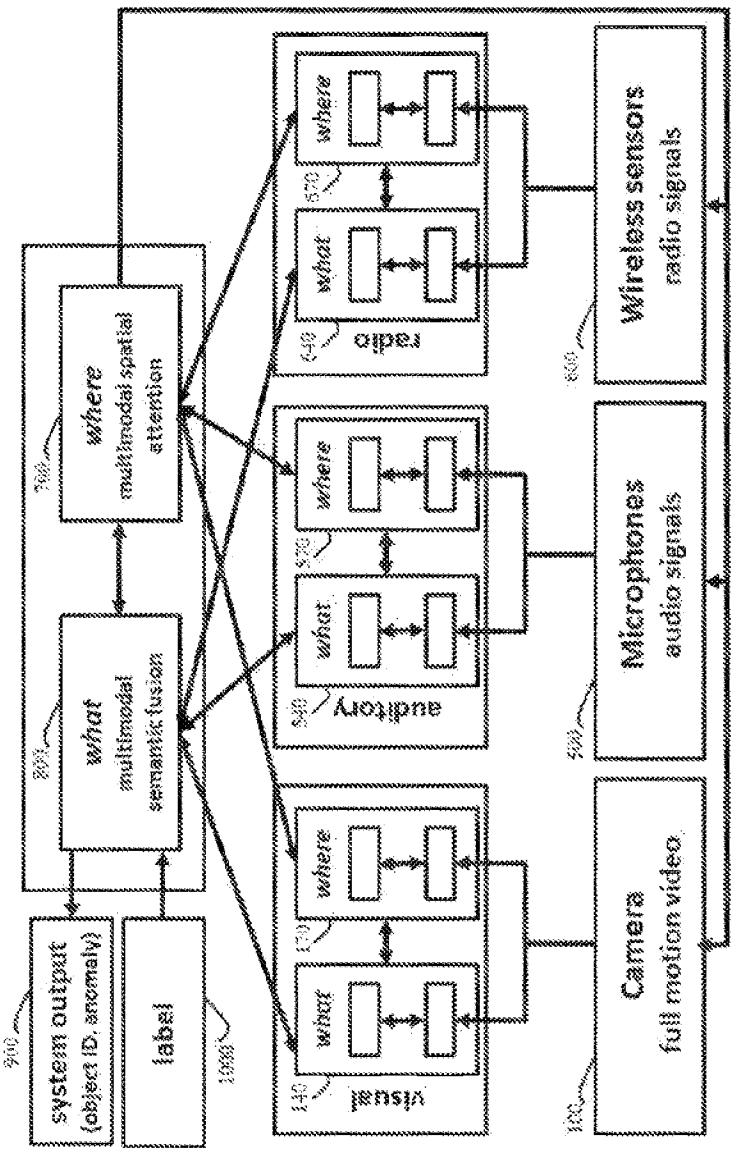

system output
(object ID, anomaly)
label
what
multimodal
semantic fusion
where
multimodal spatial
attention
visual
auditory
radio
what
where
Camera
full motion video
Microphones
audio signals
Wireless sensors
radio signals

FIG. 1

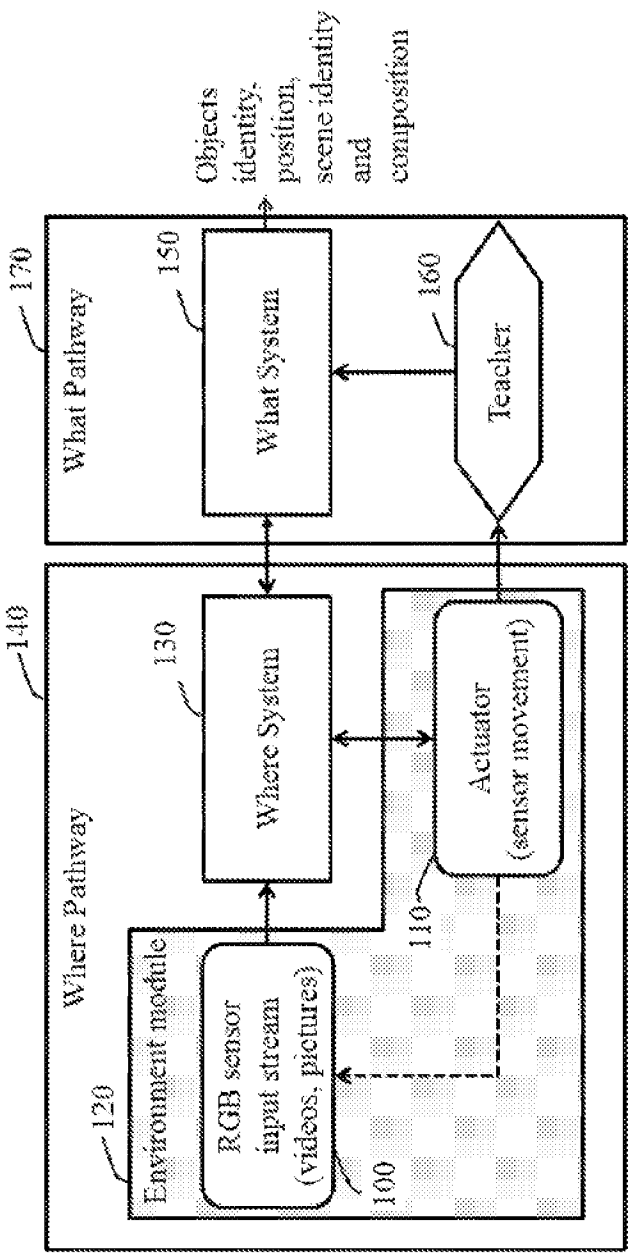

Objects identity, position, scene identity and composition
170
What Pathway
150
What System
160
Teacher
140
Where Pathway
130
Where System
Actuator (sensor movement)
110
Environment module
120
RGB sensor input stream (videos, pictures)
100

FIG. 2A

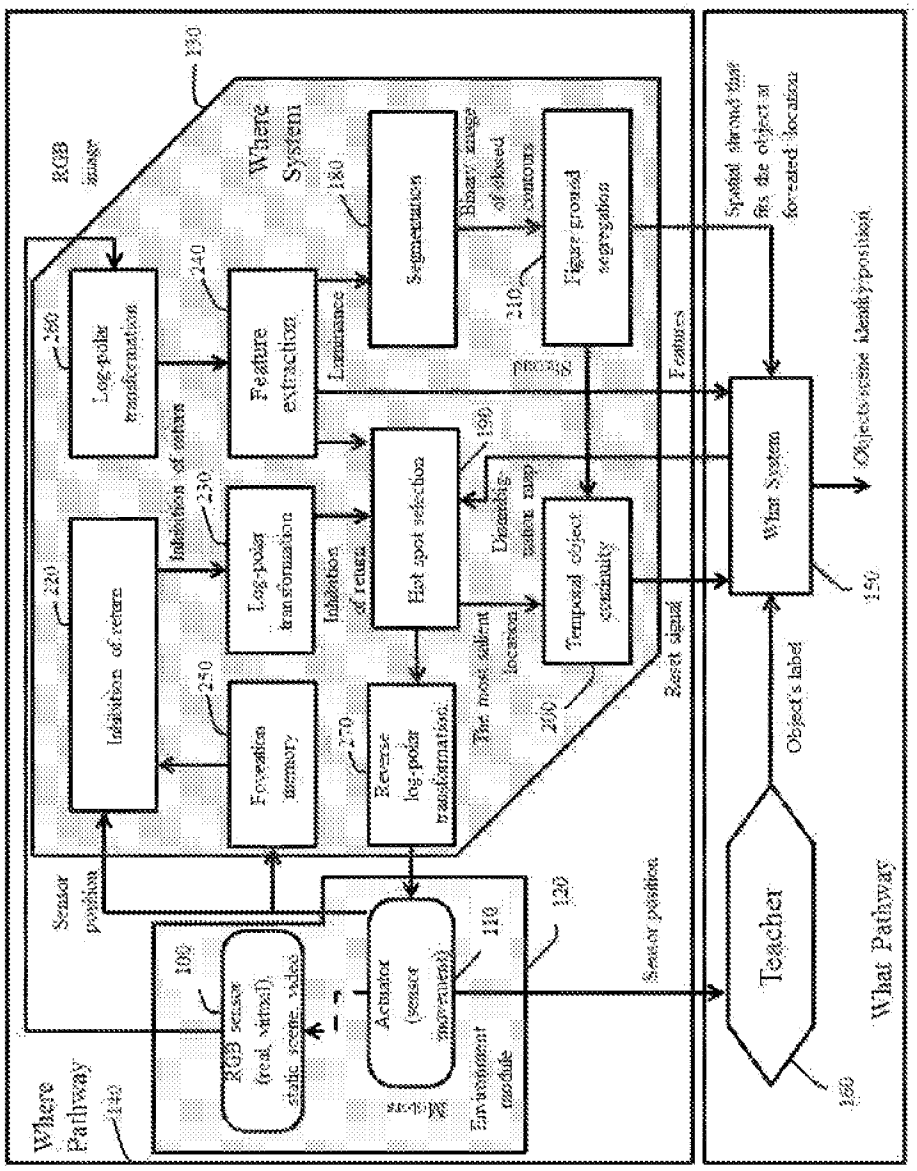

Where Pathway
What Pathway
Where System
What System
Teacher
RGB image
Log-polar transformation
Feature extraction
Segmentation
Figure ground segregation
Inhibition of return
Log-polar transformation
Hot spot selection
Temporal object continuity
Foveation memory
Reverse log-polar transformation
Actuator (sensor movement)
Environment/ database
Sensor position
Object's label
Object's name/ identity/position
Features
Binary image of filled contours
Learning/ salient map
The most salient location
Reset signal

FIG. 2B

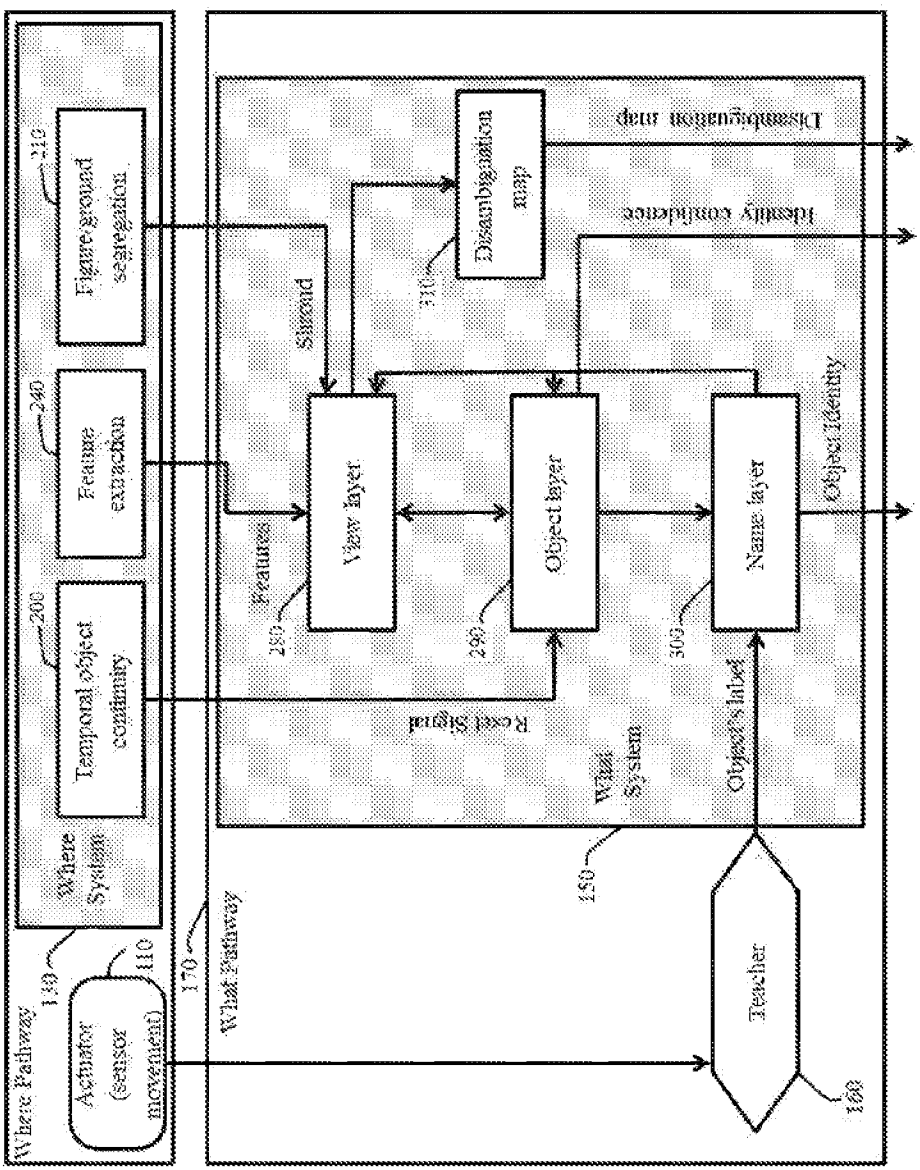

Where Pathway
Actuator (sensor movement)
110
Where System
149
Temporal object continuity
200
Feature extraction
240
Figure-ground segregation
210
What Pathway
170
What System
150
Reset Signal
Features
280
View layer
290
Object layer
300
Name layer
Object's label
Teacher
160
Disambiguation map
310
Silence
Identity confidence
Object Identity
Disambiguation map

FIG. 2C

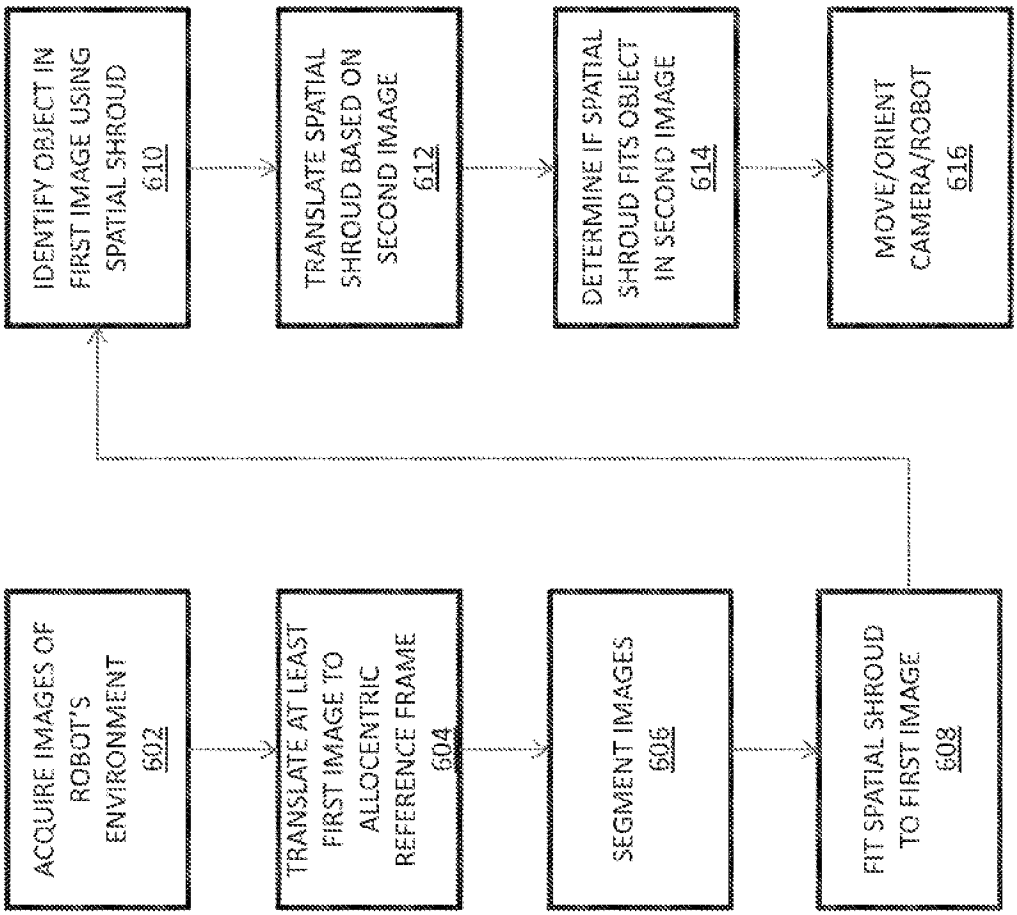
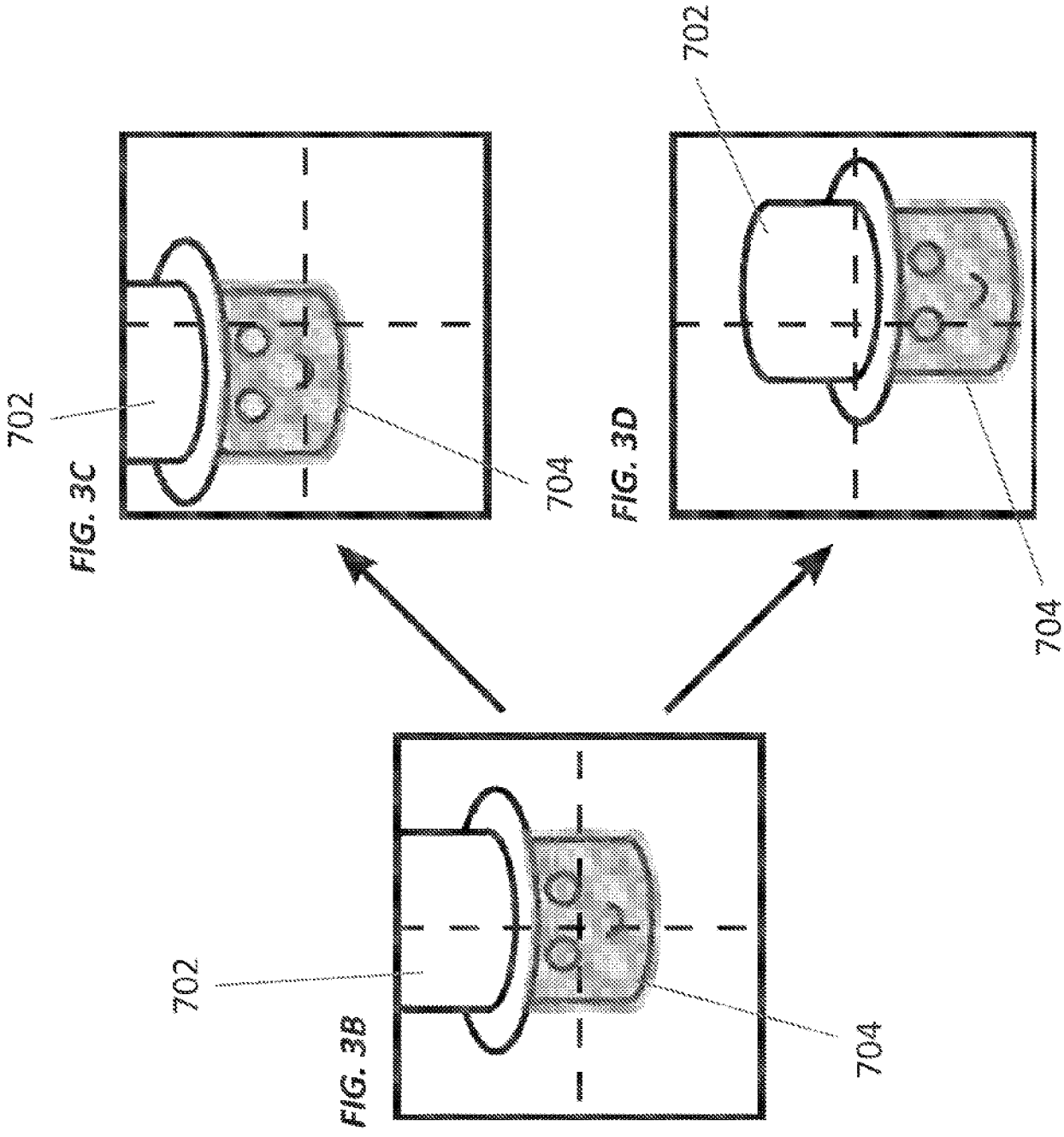

ACQUIRE IMAGES OF ROBOT'S ENVIRONMENT 602
TRANSLATE AT LEAST FIRST IMAGE TO ALLOCENTRIC REFERENCE FRAME 604
SEGMENT IMAGES 606
FIT SPATIAL SHROUD TO FIRST IMAGE 608
IDENTIFY OBJECT IN FIRST IMAGE USING SPATIAL SHROUD 610
TRANSLATE SPATIAL SHROUD BASED ON SECOND IMAGE 612
DETERMINE IF SPATIAL SHROUD FITS OBJECT IN SECOND IMAGE 614
MOVE/ORIENT CAMERA/ROBOT 616

FIG. 3A


702
704
FIG. 3C
702
704
FIG. 3D
702
704
FIG. 3B

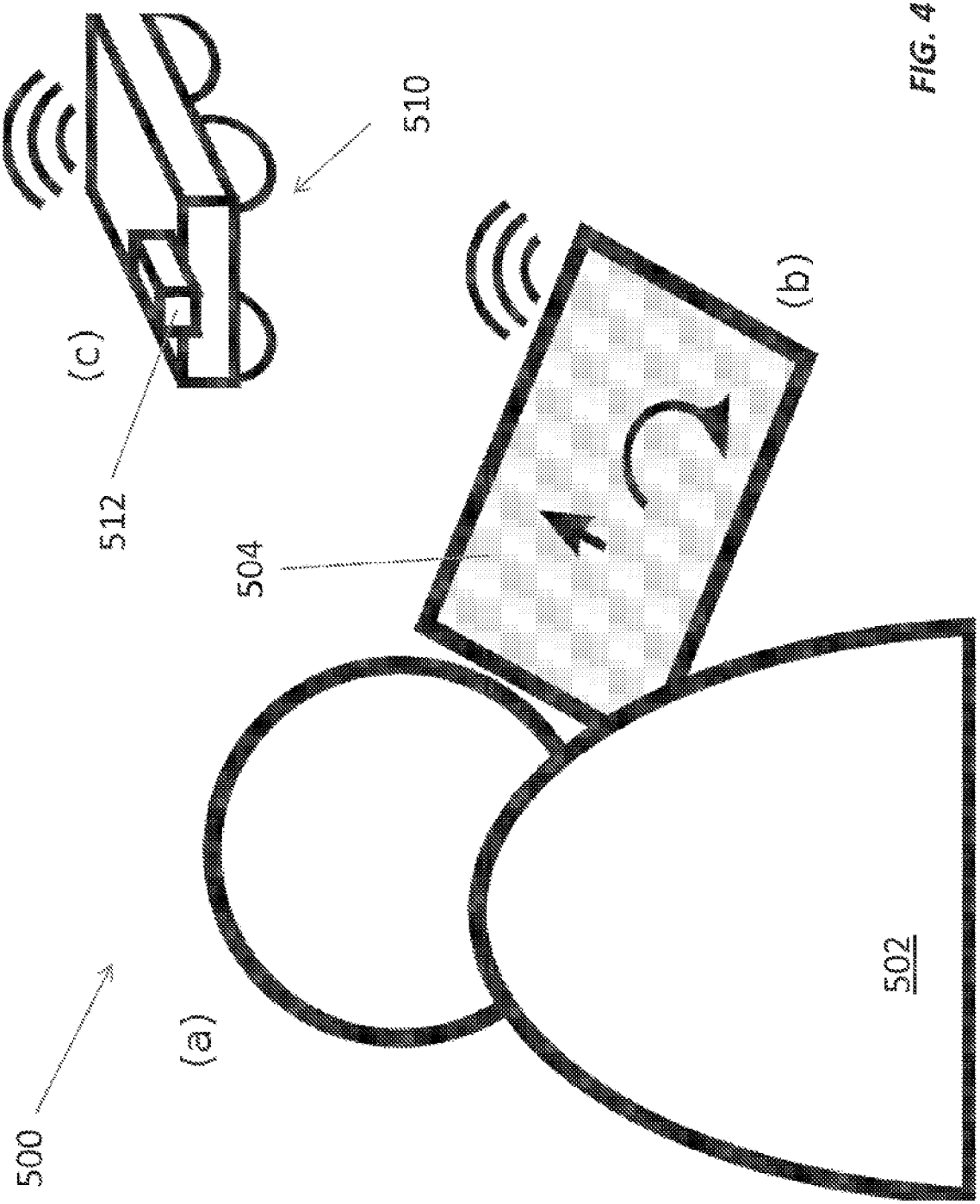


FIG. 4

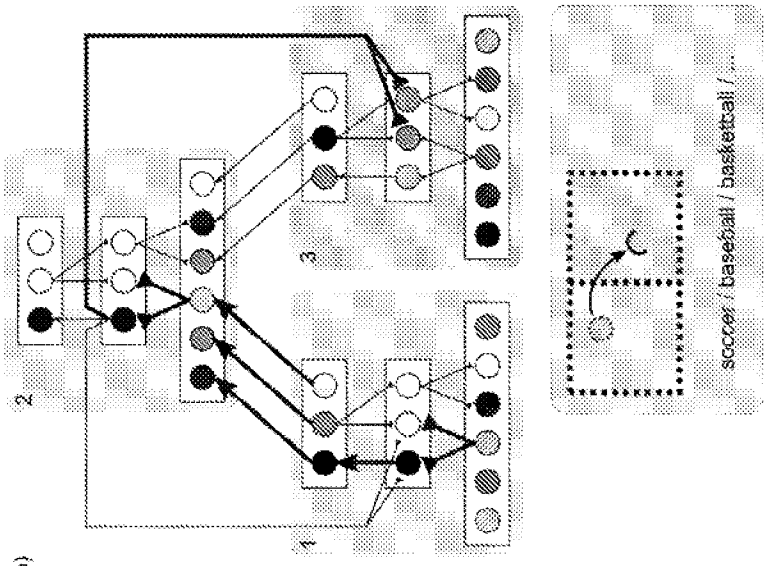

a)
Pattern 1, t0
F3
F2
F1
(i) ART search cycle
(ii) input categorized
Pattern 2, t1
(iii) F3->F2 priming
(iv) F3->F2 learns category sequence
b)
1
2
3
soccer / baseball / basketball / ...

FIG. 5A

FIG. 5B

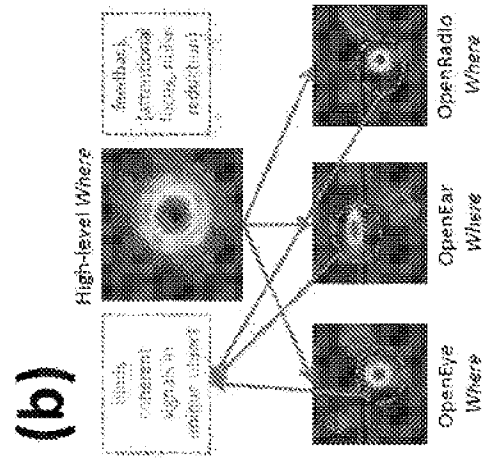

(b)
High-level Where
OpenEye Where
OpenEar Where
OpenRadio Where

FIG. 6B

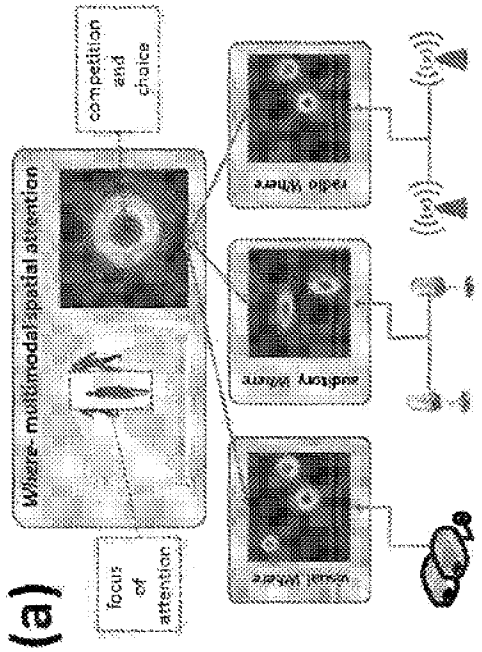

(a)
Where- multimodal spatial attention
competition and choice
focus of attention
visual filters
auditory filters
radio filters

FIG. 6A

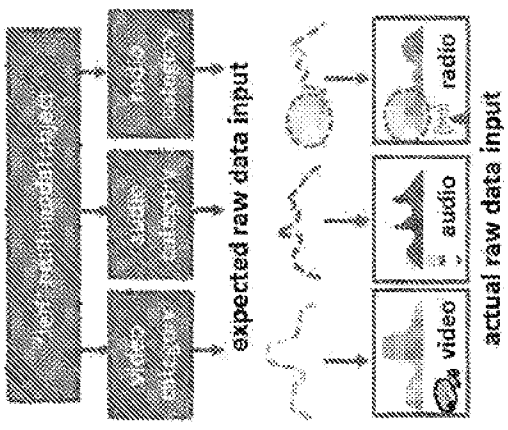

video
audio
video
audio
radio
expected raw data input
actual raw data input

FIG. 7

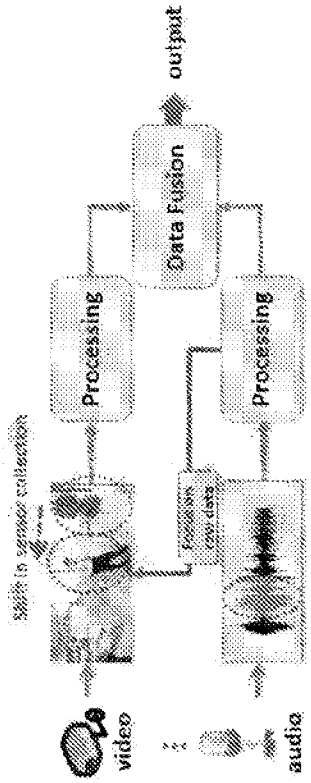

video
audio
Processing
Processing
Data Fusion
output

FIG. 8

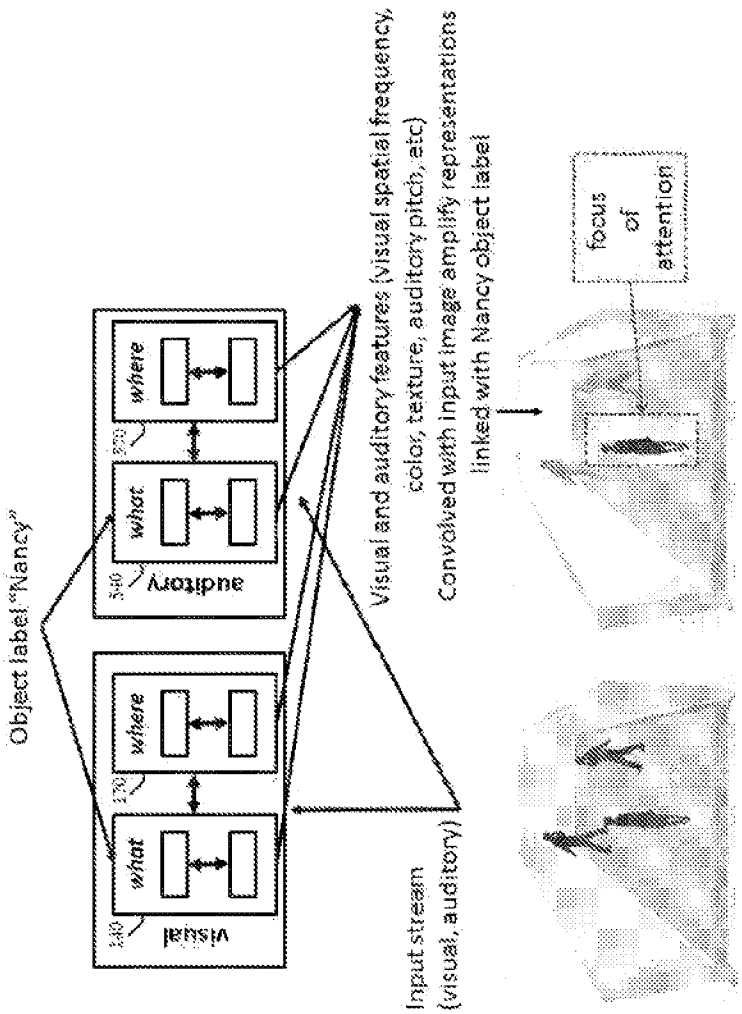

Object label "Nancy"
visual
what
where
auditory
what
where
Visual and auditory features (visual spatial frequency, color, texture, auditory pitch, etc)
Convolved with input image amplify representations linked with Nancy object label
Input stream (visual, auditory)
focus of attention

FIG. 9

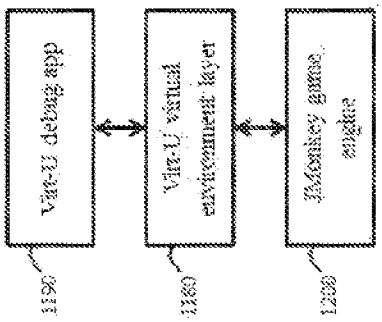


FIG. 12

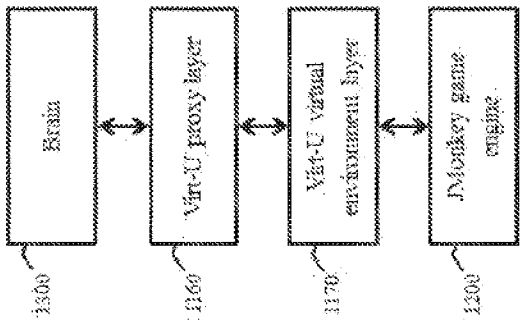


FIG. 11

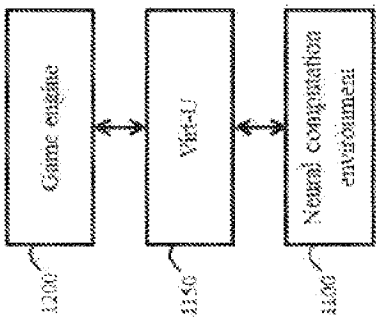


FIG. 10

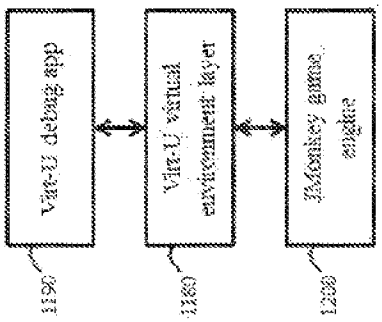


FIG. 12

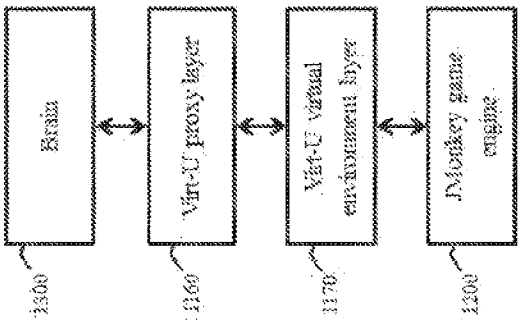


FIG. 11

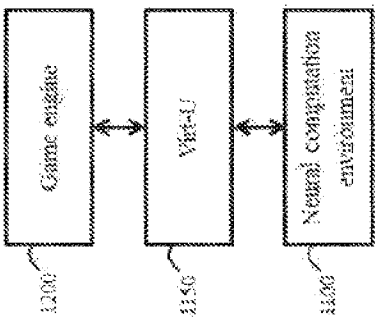


FIG. 10

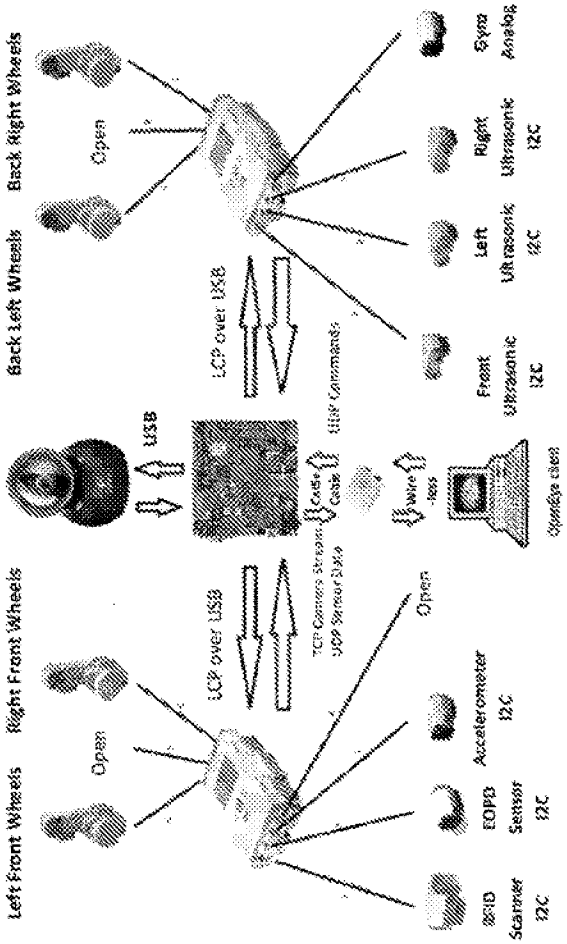

Left Front Wheels
Right Front Wheels
Open
LCP over USB
TCP Camera Stream
UDP Sensor Data
USB
Back Left Wheels
Back Right Wheels
Open
LCP over USB
UART Commands
Coffee Cable
Wave Tech
Speaker Wire
Open
RFID Scanner I2C
EOPD Sensor I2C
Accelerometer I2C
Front Ultrasonic I2C
Left Ultrasonic I2C
Right Ultrasonic I2C
Eyes Analog

FIG. 13

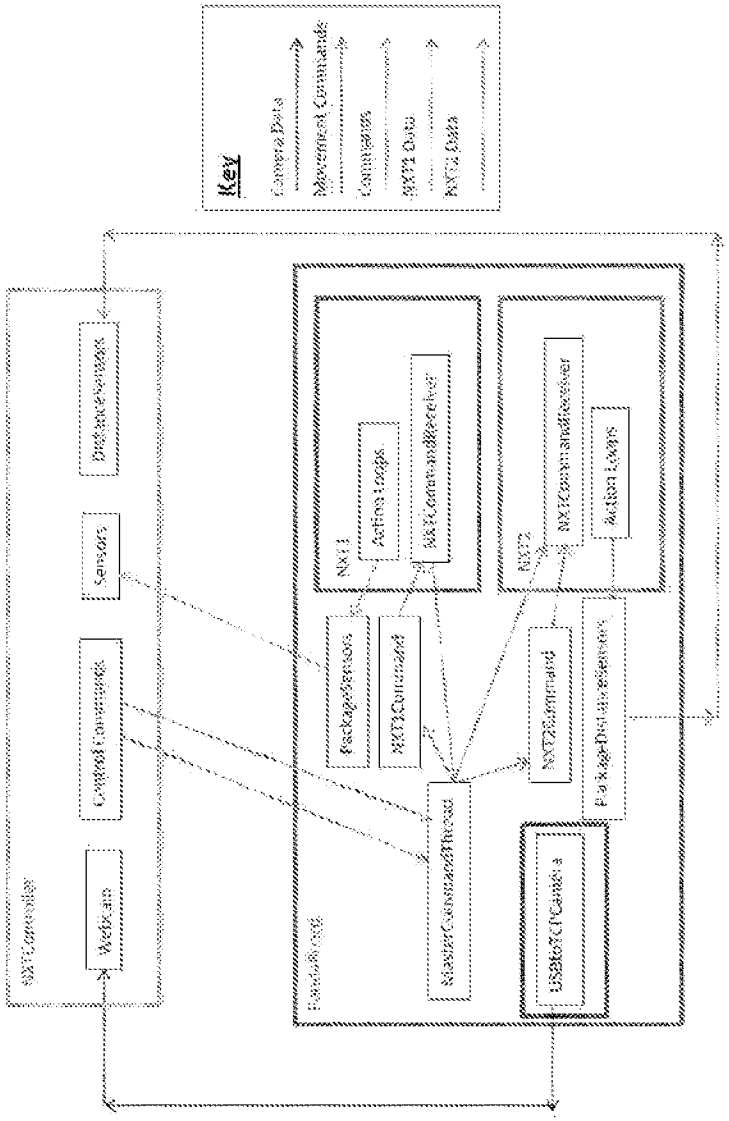

Key
Camera Data
Movement Commands
Commands
NXT1 Data
NXT2 Data
Webcam
Command Commands
Sensors
Distance Commands
NXT1
Active Loops
NXT Command Receiver
Package Sensors
NXT1 Command
Master Command Thread
NXT2
NXT Command Receiver
Action Loops
NXT2 Command
Package Distance Sensors
USB to PC Camera

FIG. 14

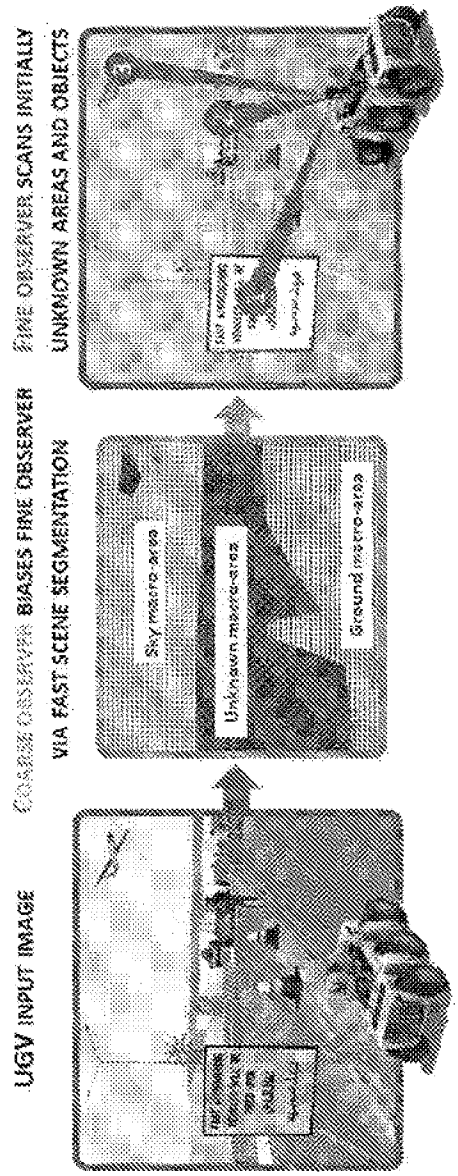

UGV INPUT IMAGE
COARSE OBSERVER BIASES FINE OBSERVER VIA FAST SCENE SEGMENTATION
FINE OBSERVER SCANS INITIALLY UNKNOWN AREAS AND OBJECTS
Sky macro-area
Unknown macro-area
Ground macro-area

FIG. 15

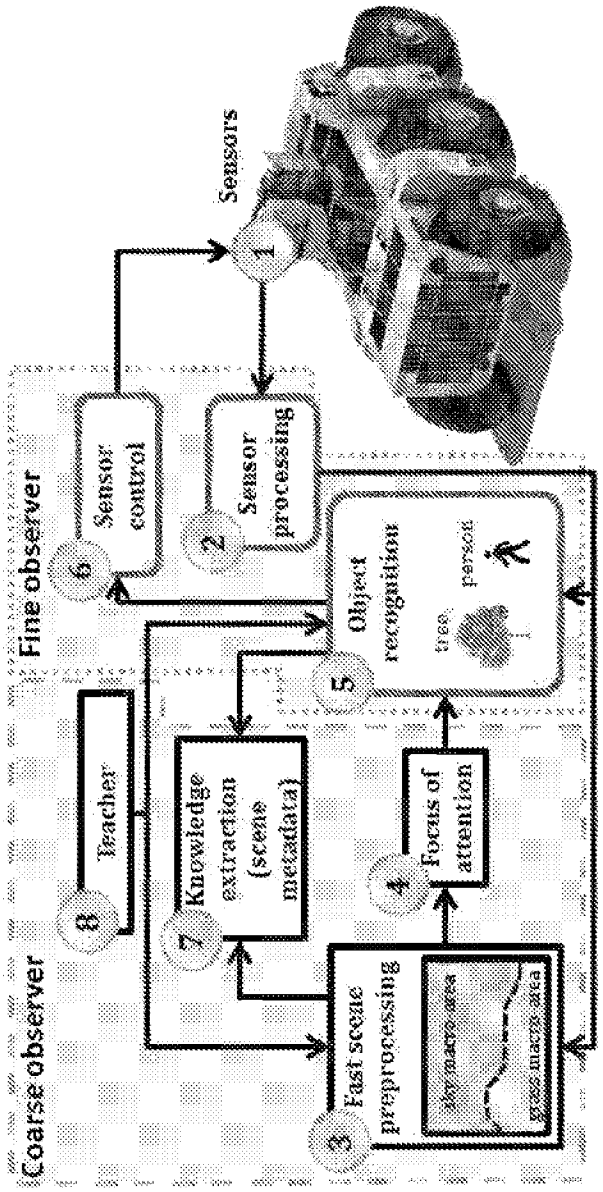

Sensors
1
Coarse observer
Fine observer
2
Sensor processing
6
Sensor control
3
Fast scene preprocessing
key macro area
coarse macro area
4
Focus of attention
5
Object recognition
tree
person
7
Knowledge extraction (scene metadata)
8
Teacher

FIG. 16

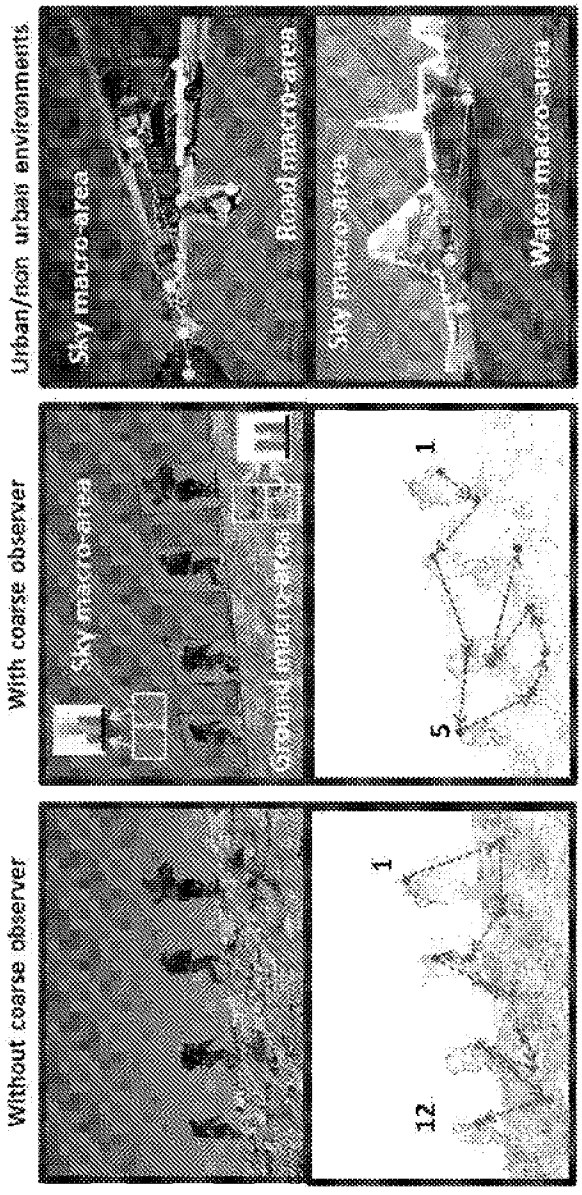

Without coarse observer
With coarse observer
Urban/non urban environments
Sky macro-area
Road macro-area
Sky macro-area
Water macro-area
1
12
1
5

FIG. 17

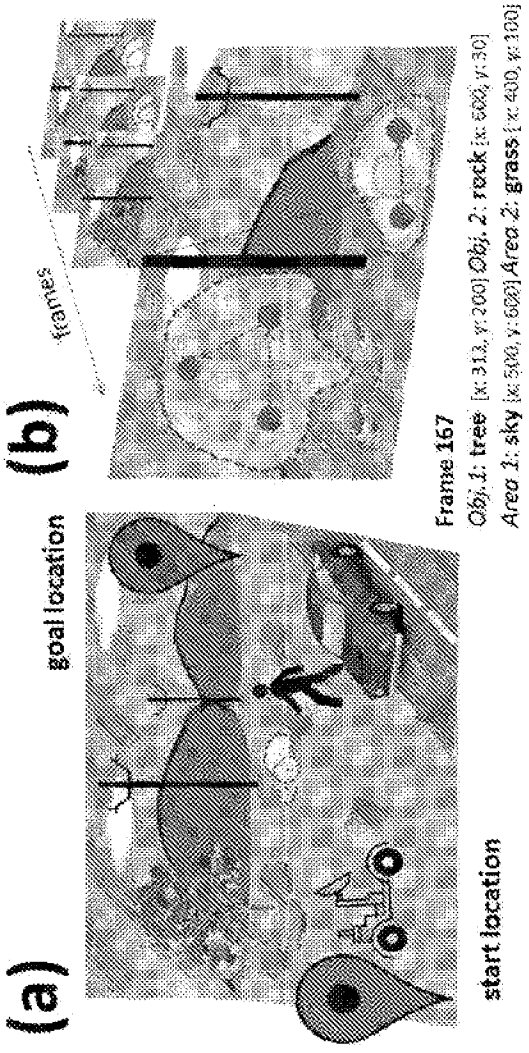


FIG. 18A

FIG. 18B

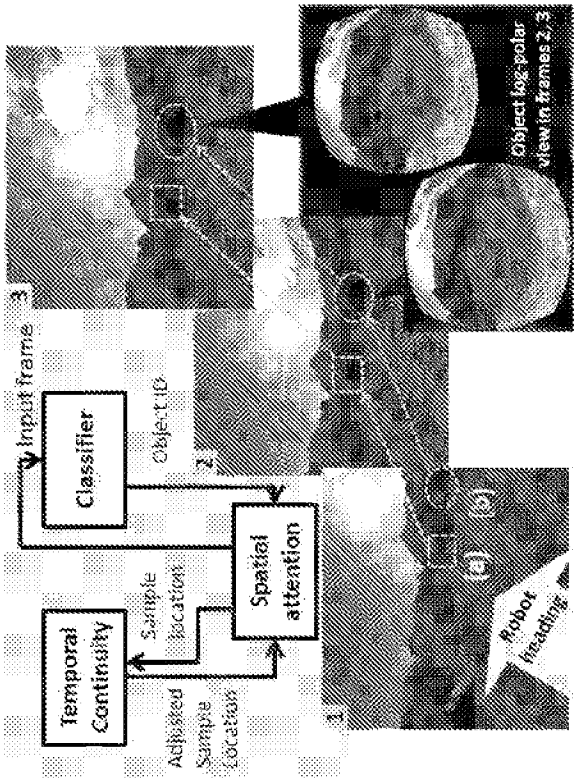

Temporal Continuity
Sample location
Adjusted Sample location
Spatial attention
Classifier
Object ID
Input frame
Robot heading
Object top-polar view in frames 2, 3
1
2
3

FIG. 19

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 January 2016 (28.01.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/014137 A2

(51) International Patent Classification:
G06F 19/00 (2011.01)

(21) International Application Number:
PCT/US2015/029438

(22) International Filing Date:
6 May 2015 (06.05.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/989,332 6 May 2014 (06.05.2014) US

(71) Applicant: **NEURALA, INC.** [US/US]; The Photonic Center, 8 St. Mary's Street, Suite 613, Boston, Massachusetts 02215 (US).

(72) Inventors; and
(71) Applicants : **GORCHETCHNIKOV, Anatoli** [US/US]; 15 Jefferson St., Newton, Massachusetts 02458 (US). **AMES, Heather Marie** [US/US]; 846 East 3rd Street, Boston, Massachusetts 02127 (US).

(72) Inventors: **VERSACE, Massimiliano**; 846 East 3rd Street, Boston, Massachusetts 02127 (US). **MATUS, Roger**; 501 Lexington Street, Apt. 15, Waltham, Massachusetts 02452 (US). **DEFREITAS, Alexandrea**; 263 Shamrock Dr., Warwick, Rhode Island 02886 (US). **AMADEO, Mike**; 7 Francestown Turnpike, Mont Vernon, New Hampshire 03057 (US). **SEEMANN, Tim**; 45 Winslow Road, Apt. 1, Brookline, Massachusetts 02446 (US). **MARSH, Ethan**; 43 Pratt Street, Apt. 1, Allston, Massachusetts 02134 (US).

(74) Agents: **TEJA, Joseph, Jr.** et al.; Cooley LLP, 1299 Pennsylvania Avenue, NW, Suite 700, Washington, District of Columbia 20004-2400 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: APPARATUSES, METHODS, AND SYSTEMS FOR DEFINING HARDWARE-AGNOSTIC BRAINS FOR AUTONOMOUS ROBOTS

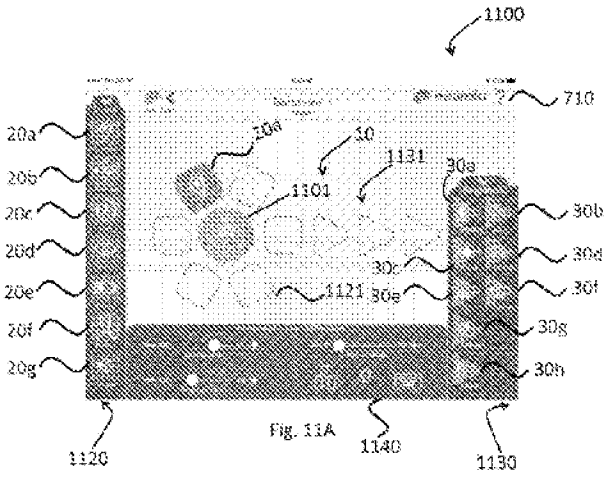


Fig. 11A

(57) Abstract: Conventionally, robots are typically either programmed to complete tasks using a programming language (either text or graphical), shown what to do for repetitive tasks, or operated remotely by a user. The present technology replaces or augments conventional robot programming and control by enabling a user to define a hardware-agnostic brain that uses Artificial Intelligence (AI) systems, machine vision systems, and neural networks to control a robot based on sensory input acquired by the robot's sensors. The interface for defining the brain allows the user to create behaviors from combinations of sensor stimuli and robot actions, or responses, and to group these behaviors to form brains. An Application Program Interface (API) underneath the interface translates the behaviors' inputs and outputs into API calls and commands specific to particular robots. This allows the user to port brains among different types of robot to robot without knowing specifics of the robot commands.

WO 2016/014137 A2

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

Apparatuses, Methods, and Systems for Defining Hardware-Agnostic Brains for Autonomous Robots

Cross-Reference to Related Application

[0001] This application claims priority, under 35 U.S.C. § 119(e), from U.S. Application No. 61/989,332, filed May 6, 2014, and entitled "Apparatuses, Methods, and Systems for Defining Hardware-Agnostic Brains for Autonomous Robots," which application is hereby incorporated herein by reference in its entirety.

Background

[0002] Conventionally, robots are typically programmed to complete tasks using a text or graphical programming language, shown what to do for repetitive tasks (e.g., as in the case of the Rethink Robotics Baxter), or operated remotely by a user. Each interface is typically specific for a type of robot or robot operating system, is often tailored to a specific task, and does not scale past its original purpose. New Artificial Intelligence (AI) techniques push the boundaries of capabilities robots can exhibit. For example, machine vision systems and, in particular, neural network (including Deep Neural Network) systems exist that provide ways to elaborate and extract actionable information from input streams in the forms, for instance, of object recognition, speech recognition, mapping (positioning of the robot in space), or ways to act upon speech and object information (e.g., by controlling motor output from robot effector/cameras/user interface/speech synthesizers).

[0003] But the programmability of conventional systems for controlling robots constitutes an issue, as they are most often seen as "black boxes" providing limited ability from the user to understand ways to harness their power for practical applications or to create usable workflows for the robot. In addition, typical/conventional robots are often programmed for specific tasks in a way that often does not scale past the original purpose. Any modifications to this programming may requires close interaction with a programmer, which can be costly and time-consuming. Other robot control solutions also take a less flexible approach to the problem at hand; for example, a shift in lighting conditions or other environmental conditions can throw off an image-recognition system that is treated like a black box.

Summary

[0004] The apparatuses, methods, and systems described herein include a new robot/platform agnostic graphical user interface and underlying software engine to allow users to create autonomous behaviors in robots and robot workflow regardless of the robot's underlying hardware and/or software/operating system. The methods are based on a stimulus-response

NVIDIA Ex1002 Page 399 of 1081

paradigm, where multiple stimuli, e.g., input from robot sensors and/or a change in a robot's state, such as but not limited to detection of a certain color or a face/object in the robot camera input image, the internal clock of the robot processor -- e.g., an iPhone or Android phone -- reaching a given time of the day, the movement of the phone as sampled by the robot accelerometer and gyro, the robot's position -- e.g., being within particular range of a given GPS coordinate, or on a location of the robot internal map, and/or similar inputs --, can trigger a response, -- e.g., an alert to the user, a pre-recorded set of movements, navigation towards a location in the robot internal map, a reaching/grasping of the robot, a synthetic speech output, and/or other actions by the robot. A special case of stimuli/responses is represented by those applications where artificial intelligence (AI) and machine vision algorithms (e.g., algorithms commonly available in software packages such as OpenCV – Open Computer Vision), such as but not limited to artificial neural networks (ANN) and their subclass Deep Neural Networks (DNN) are used to provide stimuli (e.g., visual/auditory/other sensory objects identification and classification, speech classification, spatial learning) and responses (e.g., reaching/grasping/navigation).

[0005] In certain applications, a system may be comprised of several specialized systems or networks, each dedicated to a specific aspects of perception/robot control, or can be constituted by an integrated system, namely one system/network. An example network can include the following hardware and devices, each potentially hosting a processor or group of processors: a robot, a controller (e.g., a tablet or cell phone), a local server, a cloud server, or a combination of the aforementioned possibilities. While implementations below discuss using the apparatuses, methods and systems described herein with mobile robots, such as domestic, service, and industrial robots, military robots, drones, toy robots, and the like, it should be appreciated that any device capable of observing external and/or internal phenomenon may utilize the graphic user interface and software engine described herein. For example, heating/ventilation/air conditioning (HVAC) devices, security and surveillance systems, appliances, displays, mobile devices such as smart phones and smart watches, and/or like devices may also utilize the apparatuses, methods, and systems described herein.

[0006] Embodiments of the present technology also include a graphical user interface, as embodied in a suitable computing device (e.g., a computer, tablet, or smartphone), where the user can define, select, edit, and save stimuli and responses. Coherent groups of stimuli/responses may be grouped in "behaviors". In some implementations, the user may utilize drag-and-drop interface for creating a behavior. The design may be similar in appearance to that of a biological neuron, which has a tripartite organization (the neuron body, the dendrites – where most inputs arrive – and the axon, which is the output channel of the

NVIDIA Ex1002 Page 400 of 1081

neuron). In this interpretation, dendrites are stimuli, the axon contains responses, and the cell body represents a behavior, which is the collection of stimuli and responses, but other schemes are possible.

[0007] Additional embodiments of the present technology include a method for generating a hardware-agnostic behavior of at least one electronic device, such as a robot. In one example, this method comprises receiving, from a user via a user interface executing on a computer, tablet, smartphone, etc., at least one stimulus selection corresponding to at least one stimulus detectable by the electronic device. The user interface also receives, from the user, at least one hardware-agnostic response selection that corresponds to at least one action to be performed by the electronic device in response to the stimulus. A processor coupled to the user interface generates the hardware-agnostic behavior based on the stimulus selection and the hardware-agnostic response selection.

[0008] The stimulus may come from any suitable source, including a neural network. For instance, the stimulus may comprise sensing: depressing a button; swiping a touchscreen; a change in attitude with a gyroscope; acceleration with an accelerometer; a change in battery charge; a wireless signal strength; a time of day; a date; passage of a predetermined time period; magnetic field strength; electric field strength; stress; strain; position; altitude; speed; velocity; angular velocity; trajectory; a face, object, and/or scene with an imaging detector; motion; touch; and sound and/or speech with a microphone.

[0009] Similarly, the response can be based at least in part on an output from a neural network, such as a visual object (e.g., a face) or an auditory object (e.g., a speech command) recognized by the neural network. The hardware-agnostic response selection may comprise a sequence of actions to be performed by the electronic device in response to one or more corresponding stimuli.

[0010] In some cases, this method may also include receiving, via the user interface, a selection of a particular electronic device (robot) to associate with the hardware-agnostic behavior. In response, the processor or another device may associate the hardware-agnostic behavior with the particular electronic device. The association process may involve determining identifying information for the particular electronic device, including information about at least one sensor and/or at least one actuator associated with the particular electronic device. And the processor or other device may translate the hardware-agnostic behavior into hardware-specific instructions based at least in part on this identifying information and provide the hardware-specific instructions to the particular electronic device, e.g., via a wireless communication channel (antenna).

NVIDIA Ex1002 Page 401 of 1081

[0011] If appropriate/desired, the processor may generate at least one other hardware-agnostic behavior based on at least one other stimulus selection and at least one other hardware-agnostic response selection. Possibly in response to user input, the processor may form a hardware-agnostic personality based at least on the hardware-agnostic robot behavior and at least one other hardware-agnostic robot behavior.

[0012] In another embodiment, the present technology comprises a system for generating a hardware-agnostic behavior of at least one electronic device (robot). Such a system may comprise a user interface, a processor operably coupled to the user interface, and a communications port (e.g., a wireless transceiver or wired communications port) operably coupled to the processor. In operation, the user interface receives, from a user, (i) at least one stimulus selection corresponding to at least one stimulus detectable by the electronic device and (ii) at least one hardware-agnostic response selection corresponding to at least one action to be performed by the electronic device in response to the stimulus. The processor generates the hardware-agnostic behavior based on the stimulus selection and the hardware-agnostic response selection. And the communications port provides the hardware-agnostic behavior to the electronic device.

[0013] The system may also include a hardware translation component (e.g., an Application Program Interface (API)) that is operably coupled to the communications port and/or to the processor. In operation, the hardware translation component translates the hardware-agnostic behavior into a set of hardware-specific input triggers to be sensed by the electronic device and a set of hardware-specific actions in response to the set of hardware-specific input triggers to be performed by the electronic device.

[0014] Yet another embodiment of the present technology comprises a computer-implemented method for loading at least one hardware-agnostic behavior between a first robot and a second robot. One example of this method comprises: receiving a request (e.g., via a user interface) to load a first hardware-agnostic behavior onto the first robot; retrieving the first hardware-agnostic behavior from at least one storage device, where the first hardware-agnostic behavior defines at least one first hardware-agnostic robot response to at least one first hardware-agnostic robot sensor stimulus; providing the first hardware-agnostic behavior to the first robot (e.g., via a wireless connection); providing the first hardware-agnostic behavior to the second robot (e.g., via the wireless connection); receiving a request to load a second hardware-agnostic behavior onto the first robot (e.g., via the user interface), where the second hardware-agnostic behavior defines at least one second hardware-agnostic robot response to at least one second hardware-agnostic robot sensor stimulus; retrieving the second hardware-

NVIDIA Ex1002 Page 402 of 1081

agnostic behavior from the at least one storage device; and providing the second hardware-agnostic behavior to the first robot (e.g., via the wireless connection). For example, in providing the second hardware-agnostic behavior to the first robot, the first hardware-agnostic behavior may be replaced with the second hardware-agnostic behavior. In some cases, this method may also include providing the second hardware-agnostic behavior to the second robot.

[0015] Still another embodiment of the present technology comprises a computer-implemented method for generating behaviors for a robot. An example of this method comprises receiving, at a user interface, a selection of at least one stimulus to be sensed by the robot and a selection of at least one response to be performed by the robot, e.g., in response to the selected stimulus. One or more processors operably coupled to the user interface generates a behavior for the robot based at least in part on the stimulus and the response and renders, via the user interface, the behavior as a behavior neuron. This behavior neuron may appear as a dendrite that represents the stimulus and at least part of a neuron axon (e.g., a myelin sheath section of the axon) representing the response. In some cases, the behavior neuron may be rendered as one neuron in a plurality of neurons in a graphical representation of a brain. For instance, the graphical representation of the brain may show the neuron based on the nature of the behavior in relation to behavior centers of an animal brain.

[0016] And another embodiment of the present technology comprises a method of engaging at least one hardware-agnostic behavior to control at least one robot. The hardware-agnostic behavior comprises at least one action to be performed by the robot in response to the stimulus sensed by the robot. In at least one example, this method comprises establishing a communications connection between the robot and a graphical user interface (GUI). The GUI receives an indication from a user regarding selection of the hardware-agnostic behavior. A processor or other suitable device coupled to the GUI retrieves, from a memory operably coupled to the control device, instructions for causing the robot to operate according to the hardware-agnostic behavior. The processor executes the instructions so as to engage the hardware-agnostic behavior to control the robot.

[0017] It should be appreciated that all combinations of the foregoing concepts and additional concepts discussed in greater detail below (provided such concepts are not mutually inconsistent) are contemplated as being part of the inventive subject matter disclosed herein. In particular, all combinations of claimed subject matter appearing at the end of this disclosure are contemplated as being part of the inventive subject matter disclosed herein. It should also be appreciated that terminology explicitly employed herein that also may appear in any

NVIDIA Ex1002 Page 403 of 1081

disclosure incorporated by reference should be accorded a meaning most consistent with the particular concepts disclosed herein.

Brief Description of the Drawings

[0018] The skilled artisan will understand that the drawings primarily are for illustrative purposes and are not intended to limit the scope of the inventive subject matter described herein. The drawings are not necessarily to scale; in some instances, various aspects of the inventive subject matter disclosed herein may be shown exaggerated or enlarged in the drawings to facilitate an understanding of different features. In the drawings, like reference characters generally refer to like features (e.g., functionally similar and/or structurally similar elements).

[0019] FIG. 1 shows a schematic block diagram of a system for defining and executing a Hardware-Agnostic Brain System for Autonomous Robots.

[0020] FIG. 2 is an illustration of possible connectivity schemes between controlling device where the user interface runs, the robot, and other intermediates connectivity steps, including relationship with cloud computing devices.

[0021] FIG. 3A illustrates an example implementation of a Behavior (shown as circle) as a collection of Stimuli (squares) and Responses (triangles).

[0022] FIG. 3B depicts a biological neuron, which has a structure analogous to the behavior implementation shown in FIG. 3A.

[0023] FIG. 4A shows a schematic representation of the collection of Behaviors as Brain.

[0024] FIG. 4B shows a schematic representation of different Brains as having different sets of Behaviors.

[0025] FIG. 5A illustrates variations in Brains, with each Brain comprising its unique constituent Behaviors (circles), and each Behavior comprising a unique set of Stimuli (squares) and Responses (triangles).

[0026] FIG. 5B illustrates how an output (response) from one behavior (neuron) can be used as a stimulus for another behavior.

[0027] FIG. 6 shows an example of a drone with 2 Behaviors: Behavior 1 (1 stimulus, 1 response) and Behavior 2 (2 stimuli, 2 responses).

NVIDIA Ex1002 Page 404 of 1081

[0028] FIG. 7 shows a welcome screen for an exemplary Graphical User Interface (GUI) for defining a hardware-agnostic brain and controlling a robot using a hardware/robot-agnostic interface.

[0029] FIGS. 8A and 8B show a robot selection screen in the GUI of FIG. 7.

[0030] FIGS. 9A and 9B show various exemplary GUI screens to select, add, delete, and save customized Brains using the GUI of FIG. 7.

[0031] FIGS. 10A–10C show various exemplary GUI screens to edit and configure customized Brains using the GUI of FIG. 7.

[0032] FIGS. 11A–11E show various exemplary GUI screens to configure and save customized Behaviors by choosing, adding, deleting various Stimuli and Responses using the GUI of FIG. 7.

[0033] FIGS. 12A–12D show various exemplary GUI screens with operational views of controlling a robot with a hardware-agnostic brain.

[0034] FIG. 13 shows an exemplary GUI of a Robot Knowledge Center that the User can use to label knowledge learned or collected by the robot's sensor network.

[0035] FIG. 14 shows an exemplary embodiment of a construction process of a generalized Application Program Interface (API) for defining and implementing a hardware-agnostic brain.

[0036] FIG. 15A describes an exemplary reactionary process flow of a Robot configured with a Hardware-Agnostic Brain with "n" number of Behaviors.

[0037] FIG. 15B is a flow diagram that illustrates a scheduling process executed by the in FIG. 15A.

Detailed Description

[0038] Robots are typically programmed to perform specific functions using a programming language, typically to execute repetitive tasks or to be operated remotely by a user. For many robots, each mode of operation and/or interface is specific for a particular type of robot or robot operating system. Many commercially available autonomous robots of various types or forms are pre-programmed in the programming language of manufacturer's choosing. Deploying a plurality of dissimilar robots in a centralized fashion by a single user would thus require learning and utilizing many of robots' hardware-specific "native" programming languages. This makes it impractical for a user with limited time and/or programming experience to program and use more than one type of robot because the user would have to configure each type of robot individually. In other words, a user would need to craft one specific set of actionable

NVIDIA Ex1002 Page 405 of 1081

instructions for one type of robot, for example, for general crop surveillance, another instruction set for another type of robot, for example, for close-up spotting of irrigation and/or pest problems, and another instruction set for another type of robot, for example, to spray pesticides to the affected regions of the crop.

[0039] Benefits and advantages of the present technology include, but are not limited to simplicity of use, no need for technical expertise, and scalability. By way of simplicity of use, rather than thinking in terms of sensor readings, thresholds, and continually running algorithms, the user can design robot workflow based on discrete stimuli and responses to stimuli (e.g., when the robot senses something, it performs a particular action). In addition, the present technology does not require programming lines because it supports work done through a Graphic User Interface (GUI), making it accessible to non-technical users. And solutions scale from one robot to another and can use the same interface.

[0040] Platforms for Defining and Implementing Hardware-Agnostic Brains

[0041] FIG. 1 illustrates a platform 100 that can be used to define and implement a hardware-agnostic brain for controlling almost any robot 106. The platform 100 addresses the fragmentation problem with platform and hardware-specific programming languages identified above. It also provides an intuitive, easy-to-use interface for programming robots. And it allows a user to port and apply a preprogrammed behavior or set of behaviors among many robots, including dissimilar robots.

[0042] The platform 100 includes a user interface 102 that enables a user to define a hardware-agnostic brain, a processor 104 that implements the hardware-agnostic brain (which may include processes and programs implementing Artificial Intelligence (AI)/Artificial Neural Network (ANN)/Deep Neural Network (DNN) processing), a memory 103 to store instructions for defining and executing the hardware-agnostic brain (including instructions implementing AI/ANN/DNN and synaptic weights defining ANN/DNN structures), and a communications interface 105 for communicating with the robot 106. The user interface 102 allows the user to create actionable tasks and/or usable workflows for the robot 106. The platform 100 interprets and implements these workflows as a hardware-agnostic brain 104 that interprets data from the robot 106 and input entered via the user interface 102, then performs one or more corresponding actions. The platform 100 can be implemented in any suitable computing device, including but not limited to a tablet computer (e.g., an iPad), a smartphone, a single-board computer, a desktop computer, a laptop, either local or in the cloud, etc. The platform 100 may provide the user interface 102 as a Graphical User Interface (GUI) via a touchscreen or other suitable display.

NVIDIA Ex1002 Page 406 of 1081

[0043] The user interface 102 includes a single GUI to run an underlying Application Programming Interface (API) for interfacing with a hardware-agnostic brain 104 and for communicating with and controlling the robot 106. The GUI for the user interface 102, for example, may include any shape or form of graphical or text-based programmable keys or buttons to input instructions and commands to configure the brain 104 via a touchscreen of an iPad, an Android tablet, or any suitable computing device with interactive input capabilities. A user, such as a non-technical farmer, can communicate and/or control any type, form or number of robots 106 by pre-programming the brain 104 using the simple user interface 102.

[0044] Brain 104 can be hardware-agnostic in that it can be programmed by a user with limited time and/or programming experience to control and configure any robot 106 via a user interface 102. Hardware-agnostic brain 104 can be one, or a combination, of modern AI systems, machine vision systems, and/or in particular, neural networks (including ANNs and DNNs) systems that can provide a more complex way to elaborate and extract actionable information from input streams in the forms of, for instance, object recognition, speech recognition, mapping (positioning of the robot in space) and/or ways to act upon that information (e.g., by ways of controlling motor output from robot effector/cameras/user interface/speech synthesizers). For example, the user can create a single or combination of hardware-agnostic brain or brains 104 to configure and control any type, form, or number of robots 106.

[0045] The memory 103 serves as a storage repository and/or conduit of inputs and instructions, library and/or knowledge database (including synaptic weights of an ANN/DNN) between the user interface 102 and the hardware-agnostic brain 104. For example, one or more inputs or instructions from the user interface 102 can be stored for a specific time or duration inside the memory 103. Input information stored inside the memory 103 can also be processed and/or released to the hardware-agnostic brain 104 at a prescribed time and/or for prescribed duration. The memory 103 can also receive the input data or information from the hardware-agnostic brain 104 or from the robot 106, via the interface 105, to be stored for further processing.

[0046] The platform 100 communicates with the robot 106 via the communications interface 105, which can be a wired or, more typically, wireless interface. For instance, the interface 105 may provide a WiFi, Bluetooth, or cellular data connection to the robot 106 for sending commands and queries from the processor 104 and for relaying sensor data and query replies from the robot 106 to the processor 104. The interface 105 may also communicate with other devices, including sensors that relay information about the robot 106 or the robot's environment and computing devices, such as tablets or smartphones, that host some or all of the

NVIDIA Ex1002 Page 407 of 1081

user interface 102. The interface 105 may also communicate with a server or other computer that implements part or all of the processing for the hardware-agnostic brain.

[0047] Robot 106 can be any robot or a plurality of robots, including but not limited to wheeled robots that travel on land, walking robots that walk on any number of legs, robots that can jump or bounce, and drones, such as, for example unmanned aerial vehicles (UAVs) and unmanned underwater vehicles (UUVs). Any type, form or number of robot 106 can be programmed with hardware-agnostic brains 104 via a user interface 102 to create a universal programming platform 100 that can offer a user with limited time and/or programming experience to maximize utilization of various functionalities unique to any type, form or number of robots 106.

[0048] FIG. 2 illustrates a plurality of possible real-world scenarios of how a hardware-agnostic brain 104 can be utilized. Via a simple GUI, a user 300 can implement a hardware-agnostic brain that controls and manipulates various functions of any robot via an API without having to know and/or work in the robot's native programming language. The nature of such hardware-agnosticism of the brain can also enable controlling of and communicating with any robot independently of the program platform of the host electronics devices, communication methods, or connectivity options.

[0049] A hardware-agnostic brain can be implemented on a number of ubiquitous everyday computing devices, including but not limited to a smart phone 350, a tablet 360, a laptop computer 370, a desktop computer 380, or a server 460 via several connectivity options 400 (e.g., Wi-Fi, Bluetooth, 3G, 4G, and the like). In some cases, one or more servers 460 may process instructions and data received via a GUI provided via one or more smartphones 350 or tablets 360 and one or more robots.

[0050] By programming via the GUI of the hardware-agnostic brain on a suitable computing device, a user 300 can control one or more robots, such as a wheeled surveillance robot 480, a drone 500, and a walking toy robot 520. Some additional interaction schemes between the user and a particular robot may include longer range intermediate connectivity options that can provide wireless connections between the user 300, a robot, and any form of wireless communication nodes and platforms, such as a Wi-Fi router 430 and a cellular network tower 440, for interfacing or interconnecting with a cloud computing server or device 460.

[0051] For example, if the user 300 is employing a smart phone 350 (a) to host a user interface, the phone 350 can use one of several wireless connectivity methods 400 (e.g., Wi-Fi, Bluetooth, 3G, 4G, and the like) to connect via a wireless signal (e) to a receiver 420 in, for

NVIDIA Ex1002 Page 408 of 1081

instance, a toy robot 520. In another example, the user 300 can employ a tablet (e.g., an iPad) 360 (b) to host a user interface. The tablet 360 can use one of several wireless connectivity methods 400 (e.g., Wi-Fi, Bluetooth, 3G, 4G, and the like) to connect via a wireless signal (f) to a Wi-Fi router 430, which in turn can be connected to a receiver 420 in, for instance, a drone 500. In another example, the user 300 can employ a laptop computer 370 (c) to host a user interface. The laptop computer 370 can use one of several wireless connectivity methods 400 (e.g., Wi-Fi, Bluetooth, 3G, 4G, and the like) to connect via a RC device 435 (h) in turn connected to a receiver 420 (i) in, for instance, a drone 500.

[0052] In another example, a user 300 can employ a desktop computer 380 (d) to host a user interface. The desktop computer 380 can use one of several wireless connectivity methods 400 (e.g., Wi-Fi, Bluetooth, 3G, 4G, and the like) to connect via a wireless signal (l) to a cellular network tower 440, which in turn can be connected to a receiver 420 (m) in, for instance, a wheeled surveillance robot 480. The desktop computer 380 (d) can also use one of several wireless connectivity methods 400 (e.g., Wi-Fi, Bluetooth, 3G, 4G, and the like) to connect via a wireless signal (n) to a computing (e.g., cloud) server 460, which in turn can be connected to a Wi-Fi router 430 (o), which in turn can be connected to a receiver 420 (g) in, for instance, a drone 500.

[0053] Hardware-Agnostic Robot Brains, Behaviors, Stimuli, and Responses

[0054] FIGS. 3A–6 illustrate fundamental building blocks for defining hardware-agnostic robot brains. As described in greater detail below, each brain comprises one or more behaviors. Each behavior, in turn, comprises one or more stimuli and one or more responses. The stimuli may represent changes in environmental parameters that the robot can detect with on-board or networked sensors; the status of the robot's internal and external components; and commands and signals received from other sources. The responses represent actions that the robot can take. Together, the stimuli and actions that make up a particular behavior specify how the robot works and functions.

[0055] FIG. 3A shows an example implementation of a behavior 10 in which stimuli 20 are represented as squares, responses 30 as triangles, and behaviors 10 (collections of stimuli and responses) are represented as circles. The representations in FIG. 3A mimic the structure of a biological neuron, which includes a cell body, dendrites, and an axon that splits into axon terminals as shown in FIG. 3B. The axon splits into multiple axon terminals. Other representations of behaviors 10, stimuli 20, and responses 30 are also possible.

NVIDIA Ex1002 Page 409 of 1081

[0056] As shown in FIG. 3A, a behavior 10 may comprise at least one stimulus 20 and at least one response 30, but can comprise a plurality of stimuli 10 and a plurality of responses 30. In some implementations, a behavior 10 may comprise a set of external responses 20 (e.g., movement of the robot via activating a motor) and/or internal responses 20 (e.g., change in the state of the robot which is not manifested externally to a human user) triggered by one or more external stimuli 30 (e.g., visual or auditory stimuli as detected by appropriate sensors, and/or elaborated by dedicated or integrated artificial intelligence (AI) systems, e.g. neural networks for perception, planning, and navigation) or internal stimuli 30 (e.g., a clock reaching a particular time, a counter reaching a particular count, the robot entering a particular state -- e.g., low battery, detection of an object in the robot camera, location of the robot in a particular location of its internal map, etc.). To trigger a response 30 in a behavior 10, all stimuli 20 may have to occur simultaneously (logical AND statement) or in a particular sequence, and when their conditions are all achieved, all responses 30 may be executed sequentially as arranged by the user (FIG. 3A).

[0057] FIGS. 4A and 4B show that a collection of behaviors 10 so defined constitutes a "brain" 40 A brain and its constituent behaviors 10 can be implemented in any suitable manner, including as a deep neural network running locally on a computer, tablet, or smartphone or remotely on a server. The brain may be stored locally on the electronic device used to define the brain, on a robot, and/or in a cloud-based storage system (a server). In some implementations the cloud computing and storage system may provide the brain to the robot; in other implementations the electronic device (e.g., mobile phone, tablet, a single-board computer on board of the robot, and/or the like) may provide the brain to the robot. In another implementation, a combination of local (on the robot), controller (e.g., a user cell phone, tablet, laptop, or desktop), and cloud (e.g., a remote server) jointly provide computing power for the brain.

[0058] In some implementations, a robot brain may also include one or more robot personalities 45, each of which may comprise one or more behaviors (e.g., a brain may include a personality comprising four behaviors, and/or the like) as shown in FIG. 4B. For example, a subset of the collection of behaviors (e.g., a subset attributed to certain types of stimuli, certain types of responses, and/or the like) may be associated with each other, forming a personality component of the brain. A brain may comprise a plurality of personality and behavior components. Personality components may allow for a faster and more streamlined process of adding and/or removing sets of behaviors in a robot brain, and/or the like, and could be used to build "brains" out of "personalities". Examples of personalities include, but are not limited to: (1) a "friendly" personality, composed of the following two behaviors: ● = dance when a person

NVIDIA Ex1002 Page 410 of 1081

is seen; ● = ask "hello, what is your name" when a person is seen; and (2) a "suspicious" personality, composed of the following three behaviors: ● = take a picture when a person is seen; ● = say "who is there" when a person is seen; ● = follow a person which enters the camera focus.

[0059] In some implementations, the robot brain may be independent from the robot's model or operating system. For example, the brain may convert a response instructing the robot to move diagonally 2 feet after seeing a person into appropriate motor commands for different robots. In some cases, the brain may convert the movement instructions into an instruction to roll forward and then turn to reach a destination for a ground-based robot using an Application Program Interface (API) provided by the robot manufacturer as described in greater detail below with respect to FIG. 7. But for a flying robot, the brain may convert the movement instruction into a set of instructions: take off, fly diagonally 2 feet, and then land, again using the API calls provided by the robot manufacturer.

[0060] FIG. 5A shows that a robot brain may comprise a hierarchy of brain components. As in FIG. 4, each behavior 10 component in FIG. 5A may comprise a set of stimuli-response components. In FIG. 5A, each stimulus in the set of stimuli is represented as a square (20); each response in the set of sequential responses is a triangle (30). Each behavior component (represented as a circle) can comprise a different number of stimuli and/or responses in the stimuli-response set.

[0061] FIG. 5A also shows that the behaviors in a brain may be given a priority order of execution. For example, if two behaviors are activated by the sensors, a priority may be assigned to the one behavior over the other behavior, e.g., based on the behaviors' positions in the brain representation. Priority behaviors may be allowed to complete a sequence of responses or not, depending on the implementation, before other behaviors are executed. Behaviors of lower priority may be interrupted before they complete execution if a higher priority behavior is activated. In other implementations, the last response in a chain of responses needs to be completed for other behaviors to be engaged.

[0062] FIG. 5B shows an example of how behaviors (neurons) in a brain can be connected to each other or "chained together." In this case, a first behavior 510a produces a particular output (e.g., motion in a particular direction or to particular GPS coordinates) that serves as a stimulus 520a for a second behavior 510b. If the second behavior's other stimulus 520b (e.g., recognition of a particular object in imagery acquired by a camera) is present, then the second behavior 510b produces its own response 530, which in turn may stimulate another behavior. Thus, triggering of the first behavior 510a is a stimulus for a second behavior 510b.

NVIDIA Ex1002 Page 411 of 1081

[0063] Additionally, behaviors can be "chained", namely, they can be organized in sequences where the stimulus for a given behavior can be represented by the completion or termination of another behavior. For example, FIG. 6 illustrates chained behaviors that are executed by a drone (UAV) to monitor a particular area

● 1= track when you see a new person
■ = an unknown person is seen by the camera
■ = it is between 5:00PM and 8:00AM
▶ = engage tracker centered around a person;
▶ = send an email notification.
● 2= survey particular Global Positioning System (GPS) coordinates at a specified time
■ = ● 1 is terminated (tracking has suspended, e.g., the person is out of sight)
■ = it is between 5:00PM and 8:00AM
▶ = survey GPS coordinates

[0064] In some implementations, a stimulus can be determined by the user performing an action on the controller. For instance, a user may select an object shown on a touchscreen or other device that displays imagery from an image sensor mounted on the robot. This selection may trigger a response by the robot brain in the form of a machine vision/AI/ANN/DNN visual learning and tracking algorithm that tracks the selected object/person.

[0065] In some implementations, even though each stimulus and response is independent of hardware on any particular robot useable via the interface, certain stimuli may not be observable by a particular robot, and/or certain responses may not be performable by a particular robot, rendering a behavior difficult and/or impossible to act upon for a particular robot. In such instances, the robot may ignore the behavior until its capabilities have changed, the user may be prompted to change the behavior when selecting a robot to use the behavior and/or a brain with the behavior, and/or the like.

[0066] Additionally, the computing medium where behaviors and their collections may be implemented may be a processor on board of the robot, the robot controller (e.g., a laptop, a tablet, or a cell phone), a remote (cloud) server, or may be partially implemented in any of the above-mentioned devices, e.g., as described above with respect to FIG. 2. E.g., in certain instantiations, the computation underlying simple behaviors especially these that require time-sensitive responses may be implemented in the robot, some others in the controllers, whereas some more complex and computationally intensive behaviors that do not require short latency responses may be implemented in the cloud, namely on a remote server connected to the robot via a tethered or wireless connectivity.

NVIDIA Ex1002 Page 412 of 1081

[0067] Additionally, given that brains and behaviors are implemented in a robot agnostic fashion, brains can be "moved" from robot to robot, e.g., by "copying" and "pasting" them from device to device without loss of functionality except from behaviors (stimuli/responses) which are incompatible between different platforms.

[0068] Examples of Stimuli and Responses

[0069] A hardware-agnostic brain can receive one or more stimuli from one or more of the following sources including, but not limited to the user input via the GUI, a network of sensors that reside on the robot, and algorithm output itself from the AI/ANN/DNN of the hardware-agnostic brain.

[0070] Examples of the user input stimuli include, but not limited to physical inputs, such as, for example a touch button press on the Control screen of an icon (e.g., a brain) of the GUI, a button press on Control screen of either the robot or the controlling unit, a finger or palm swipe on the touch screen (the user may be allowed to instruct the robot to memorize a pattern), a general reorientation motion, including tiling, rotating, turning over, touching, or any combination of manipulation, of the entire device, and a multi-digit input (e.g., a "pinch") on a touchscreen.

[0071] Examples of stimuli from a network of sensors from the robot include, but not limited to quantitative or status readings of one or more sensors, such as for example, battery indicator, strength in quantitative nature, presence or absence of 3G or 4G signal, strength in quantitative nature, presence or absence of Wi-Fi signal, strength in quantitative nature, presence or absence of Bluetooth signal, time, date, various of functions of stopwatch capabilities, including countdown, timer, etc., quantitative nature of acceleration in various units, quantitative nature of velocity in various units, quantitative nature of angular velocity in various units, quantitative nature of speed in various units, strength in quantitative nature of and pointing direction of magnetic field, orientation of magnetic azimuth (true north), quantitative nature of Latitude readings, quantitative nature of Longitude readings, quantitative nature of altitude readings, and course.....

[0072] Some other examples of stimuli from the robot are of visual nature (e.g. visual stimuli) and can include, but not limited to color detection, face detection, face recognition, object recognition, scene recognition, and gesture recognition.

[0073] Some other examples of stimuli from the robot are of audio nature (e.g. audio stimuli) and can include, but not limited to any loud sound, speech recognition, speaker recognition, musical or note patterns, and pre-defined auto cue (e.g., clapping of hands).

NVIDIA Ex1002 Page 413 of 1081

[0074] Some other examples of stimuli from the robot are of geographical nature (e.g. location stimuli) and can include, but not limited to location provided by GPS coordinate, location provided by internal map generated by the robot (e.g., SLAM maps), location provided by map supplied by the user, visual scene recognition (e.g., "this looks like the kitchen"), and auditory scene recognition (e.g., "this sounds like the kitchen").

[0075] Examples of stimuli that are algorithm output itself from the AI/ANN/DNN of the hardware-agnostic brain (e.g. algorithmically defined stimuli) can be any stimulus that is generated by the output of a machine vision (e.g., OpenCV), AI, ANN, or DNN algorithm that elaborates physical or any other stimuli input to the system

[0076] A hardware-agnostic brain can output one or more responses including, but are not limited to tracking an object/person identified by the algorithm (stimulus = detection of object/person) or by the user (stimulus = a "pinch" on the object in an iPad screen), executing user recorded actions, meaning anything that the robot can do is memorized, making a sound or another non-motion item, taking a picture, recording audio or video, executing intelligent motion, such as find a person or go to a location, posting picture/video on a social media, updating a database, sending an email notification, and engaging, launching, and/or communicating with another software application.

[0077] GUIs for Adding and Modifying Hardware-Agnostic Robot Brains

[0078] FIGS. 7–12D illustrate an exemplary GUI for creating, editing, and selecting a robot brain and for controlling a robot, either directly or with the brain. As explained below, the GUI may be organized into Connect, Control and Configure components with a navigation page.

[0079] FIG. 7 illustrates a navigation page 700 that shows four buttons, each with a unique icon, for connecting to a robot, configuring a robot bran, and controlling a robot. The buttons include "select robot" button 702 to select and connect to a robot, an "add brain to robot" button 704 to select a brain for the robot , a "see robot view" button 706 to see real-time (image) data provided by the robot's sensory suite and to control the robot directly, and an "edit brains" button 708 to edit the hardware-agnostic brains available for the robots. The navigation page 700 also includes a "home" button 701 that returns the user to the navigation page 700 and an information button 710 that leads the user to information/support options. This is one of many exemplary combinations of navigation page 700 that can be used in this approach. Other approaches may also enable, for example, to access robot selection, robot operation, brain selection, and brain editing in different combinations, graphic styles, and order, but with equivalent underlying functionalities.

NVIDIA Ex1002 Page 414 of 1081

[0080] Selecting and Connecting to a Robot

[0081] FIGS. 8A and 8B show a "select robot" screen 800 that implements the Connect component of the GUI and can be reached by selecting the "select robot" button 702 on the navigation page 700 shown in FIG. 7. The "select robot" screen 800 shows robots available for connection to the brain and options for identifying and connecting to other robots and for obtaining resources for the operation of a particular robot. The user can use this page to select and connect to a robot or swarm of robots via a wide-area network (WAN), a local-area network (LAN), or a short-range connection, such as Bluetooth or near field communication (NFC). The user may also manually input a specific IP address associated with a particular robot. Additionally, this GUI display screen can be used to connect the robot to additional computing resources or knowledge databases (e.g., additional processing resources, such as a "cloud brain"), to additional brains available in a local server, to other robots, to the Intranet/Internet (e.g., corporate directories, LinkedIn) for additional knowledge to use in brains (e.g., knowledge of faces/information about employees), and to a "brain store" (community developed brains, etc.).

[0082] In some cases, this framework allows connecting a single interface to multiple robots. (In other cases, a robot's API or protocol may allow only one robot to connect to a device at a time or to share data streams with other devices/robots.) For example, a single device may control multiple robots if the robots' API communication protocol(s) allow the robots to share streams and when the controlling device has enough processing power to handle processing on multiple data streams simultaneously (e.g., one video stream from each robot). The amount of processing to maintain and utilize a connection varies from robot to robot, so the total number of robots that can be connected to a single device depends on the device's processing power and the robots' processing requirements, among other things.

[0083] FIG. 8A shows that screen 800 enables the user to select a "recent" button 802 to show recently used robots, a "favorites" button 804 to see and modify a list of preselected favorite robots, a "search" button 806 to can enable searching for any robot, an "import" button 808 to import drivers and/or other software for controlling a particular robot, and a "network" button 810 to display a list of robots connected to a computer/communications network. Screen 800 also identifies specific robots by name and/or by type. In this case, the available robots include: a Parrot Sumo (button 820a), a Parrot Bebop (button 820b), a Vgo (button 820c), a DJI (button 820d), and a Parrot AR Drone (button 820e). Examples of robots that may be capable of interfacing with the application include, but are not limited to:

- Wheeled robots;

NVIDIA Ex1002 Page 415 of 1081

- Aerial drones and quadcopters, e.g., Parrot Bebop, Parrot AR Drone, 3D Robotics Iris, DJI Drones, Airwave autopilots;
- Toy robots with cameras or that accept external cameras, e.g., Parrot Sumo, Romotive Romo, WowWee RoboSapiens;
- Service/domestic/professional robots, e.g., iRobot Roomba, Vecna QC Bot, Harvest automation;
- Telepresence robots, e.g., Vgo, MantaroBot TeleMe, Anybots QB, Double Robotics, Beam+, iRobot AVA, Swivl; and
- Industrial robots, e.g., Kuka robots, ABB robots, Rethink Robotics robots.

The user may also select a mode that does involve connecting to a robot, such as a "NO-BOT" mode for use with an iPhone that is not in a robot body or a "tester" mode that may allow a user to use a virtual robot in a virtual environment. Tester mode may be useful for training a new user or for a person programming a brain without a working robot. These modes enable the application to run with no wireless connectivity (e.g., the user may be able to turn on airplane mode for testing, and may either connect to a virtual robot or instruct the user to try connecting to the robot once the user has a wireless connection with the robot).

[0084] If the user selects a particular robot, such as the Parrot Sumo as in FIG. 8B, screen 800 displays the selected robot in a selection bar 830. The GUI and associated processor connect to the selected robot and provide the "see robot view" button 706, which if selected presents sensor data from the robot. Selecting the "see robot view" button 706 also enables the user to control the robot as described below with respect to FIGS. 12A–12D.

[0085] Adding a Brain to a Robot

[0086] FIGS. 9A and 9B show a dedicated GUI display screen 900 that provides part of the "Configure" component. It appears if the user selects the "add brain to robot" button 704 on the navigation page 700. The screen 900 shows several icons representing various GUI functionalities including an "add brain" button 902 and buttons associated with previously defined brains, shown in FIGS. 9A and 9B as "888" brain 904a, "AllDisplays" brain 904b, "AudioResponse Test" brain 904c, and "Button Stimulus" brain 904d (collectively, previously defined brains 904). The previously defined brains 904 can be created and stored locally by the user or accessed or downloaded via a cloud resource, such as a "brain store" or a free sharing framework. Screen 900 also includes a search input 906 that enables to the user to search for a particular brain and/or filter brains by name, robot type, or other brain characteristic. Brains can be "swapped" via a GUI on the iOS device as described below.

NVIDIA Ex1002 Page 416 of 1081

[0087] Each brain (including each previously defined brain 904) may have an xml representation that can be shared across one or more devices (robots) simultaneously, sequentially, or both simultaneously and sequentially. For instance, a particular brain can be swapped among robots and/or transmitted to multiple robots via a GUI executing on a iOS device, Android device, or other suitable computing device.

[0088] The user can apply one brain to many robots, one brain to many different types of robots, and/or many brains to one robot via screen 900 without having to know or understand the specifics of the brain commands, the robots' capabilities, or how to program the robots. If the user selects a brain that is incompatible with the selected robot, the GUI may present a message warning of the incompatibilities. For example, if the selected robot is a ground robot and the brain includes a behavior for a UAV, such as a "Fly Up" command, the system warns the user that the brain and/or its behavior(s) has one or more incompatibilities with the selected robot.

[0089] FIG. 9B shows that the user can also navigate to screen 900 by selecting the "edit brains" button 708 on the navigation page 700. In "edit brains" mode, screen 900 gives the user the option to delete each previously defined brain 904 with respective delete buttons 952a–952c. When the user finishes editing or creating a brain, the brain can be saved and/or stored in many physical and virtual places including but not limited to the local memory of the controlling device, on the robot (provided the robot has adequate storage capability), or a cloud computing device. In the case of a telepresence robot controlled by an attached mobile device (e.g., an iOS or Android device), the brain can be stored on the attached mobile device.

[0090] GUI-Based Brain Editor

[0091] FIGS. 10A–10C show a brain editor screen 1000 of the GUI that enables the user to create or edit a brain 40. As described above with respect to FIGS. 3–5, each brain 40 include one or more behaviors 10, which in turn include one or more stimuli 20 and one or more response 30. The user can save, rename, or delete a brain 40 by selecting the appropriate button in a menu 1002 at the bottom of the brain editor screen 1000. The user can also navigate within the GUI using a menu button 1004.

[0092] The user can add a behavior 10 to the brain by clicking on an add behavior button 1010 as shown in FIG. 10A. Selecting or hovering over a particular behavior 10 may cause the GUI to display a pop-up window 1010 that shows the stimuli 20 and response(s) 30 defining the behavior 10 as shown in FIG. 10B. Behaviors 10 may be grouped to form personalities 45 and/or positioned in the brain 40 to establish weights or precedence among behaviors 10. For

NVIDIA Ex1002 Page 417 of 1081

example, positioning a behavior 10 in the upper left corner of the brain 40, as shown in FIG. 10B, may increase the behavior's weight, whereas positioning a behavior 10 in the lower right corner of the brain 40, as shown in FIG. 10C, may decrease the behavior's weight.

[0093] GUI-Based Behavior Editor

[0094] FIGS. 11A–11E show a Behavior Editor 1100 that enables viewing, adding, editing, and deleting of behaviors 10 for use in creating and editing brains 40. As shown in FIG. 11A, available stimuli 20 are represented on a stimulus panel 1120 and available responses 30 are represented on a response panel 1130 displayed on either side of the behavior 10 being created/edited. The available stimuli 20 and responses 30 may be retrieved from a library stored in a local or cloud-based memory.

[0095] To use the behavior editor 1100 to create or edit a behavior 10, the user selects a stimulus 20 by dragging it from the stimulus panel 1120 and dropping it into a stimulus input 1121 organized in a "petal" formation around a central circle "Save" button 1101, just like dendrites extending from a neuron body as shown in FIG. 3B. (Other arrangements are also possible.) Similarly, the user selects a response 30 by dragging it from the response panel 1130 and dropping it into a response input 1131 extending from the central circle "Save" button 1101, just like an axon extending from a neuron body as shown in FIG. 3B. The user can also set specific parameters for each stimulus and response using a stimulus/response editor 1140. For instance, for a location stimulus 20a, parameters may include longitude/latitude and radius (tolerance) of the GPS location as shown in FIG. 11A, a location on a specific map (e.g., Google map), a street name, or a location in a robot-generated map (unlabeled, or labeled – e.g., John's kitchen).

[0096] Stimuli can be linked by AND/OR logical conditions. Types of stimuli include but are not limited to: user input, such as touchscreen swipes, tilts, button pushes, etc.; machine vision (e.g., OpenCV), AI/ANN/DNN-related input (e.g., color, motion, face, object, and/or scene detection, robot-generated map); and quantitative sensor readings as well as device status from robot or controlling device, e.g. an iPad (e.g., WiFi signal strength and time of day). In some implementations there may be sub-dialogs for settings (e.g., at what battery level should a stimulus be activated). The setting may be displayed without the need to open the sub-dialog, or the user may open the sub-dialog for editing. Machine vision stimuli may include selection of particular colors the robot can detect to generate a response. Other implementations can include objects, people, scenes, either stored in the knowledge base of the robot, objects the user has trained the brain to recognize, objects that have been trained by other users, object learned by other robots, or knowledge bases available in cloud resources.

NVIDIA Ex1002 Page 418 of 1081

[0097] In this example, available stimuli include location 20a (e.g., GPS coordinates from a GPS receiver or coordinates from an inertial navigation unit), direction 20b (e.g., a heading from a compass or orientation from a gyroscope), time 20c (e.g., from a clock or timer), vision 20d (e.g., image data from a camera or other image sensor), battery 20e (e.g., a power supply reading from a power supply), user input 20f (e.g., from a button on the robot, the GUI, or another interface), and drone 20g (e.g., drone-specific stimuli, such as flight altitude). Additionally, other stimuli can be represented by the execution of another behavior.

[0098] As shown in FIGS. 11A–11E, responses 30 are depicted as triangles arranged sequentially (in a line) and selectable from the sliding panel 1130 on the right. Once conditions to satisfy a stimulus (or multiple stimuli, e.g., three stimuli arranged in AND statements) are met, one or more responses are triggered. These responses can be executed sequentially, and while being executed, other stimuli processing can be prevented from gaining access to other stimuli using a scheduler as explained below with respect to FIGS. 15A and 15B. In other implementations, the sequence of responses can be broken by intervening stimuli. Responses are converted from robot-agnostic to robot-specific in software as explained below with respect to FIG. 14. For example, a "Move forward for 2 meters" on a ground robot, and the same command on a drone, will result in two very different set of motor commands for a robot, which will need to be handled in software to achieve equivalent behavioral results.

[0099] Responses 10 can include changing the status of the display of the robot (when available), specific movement of the robot, sounds (e.g., speech), tilt/rotations of the robot, picture/video, turning on/off lights (e.g., LED), pausing the robot, drone-specific operations (e.g., take off). In this example, available responses include display 30a)(e.g., if the robot has a screen, it can be a picture/video/image on the screen, color, text, etc.), light 30b (e.g., turn on a light-emitting diode (LED)), move 30c (e.g., trigger a walking or rolling motor), sound 30d (e.g., record sound with a microphone or emit sound with a speaker), tilt 30e (e.g., with an appropriate tilt actuator), drone 30f (e.g., fly in a certain direction, speed, or altitude), camera 30g (e.g., acquire still or video images with an image sensor), and pause 30h (e.g., stop moving). Additionally, custom actions can be available from the cloud, an on-line store, or other users.

[0100] Additionally, responses can be controlled by an AI/ANN/DNN. For example, a response 10 may be "Go to the kitchen," where the knowledge of the spatial configuration of the environment is given by the robot mapping system (e.g., a DNN). Similarly, for the response "Find Bob", the knowledge of Bob is given by an AI/ANN/DNN system. And for the response "Grasp the can of coke", finding the object, reaching, and grasping can be given by an AI/ANN/DNN system.

NVIDIA Ex1002 Page 419 of 1081

[0101] Stimuli 20 and responses 30 can be re-arranged by dragging and dropping in the interface 1100, and a specific response can formed by the user recording specific movement by the robot performed under the control of the user, and saved as custom movements. For example, in FIGS. 11A and 11B, the user selects and adds the location stimulus 20a and the vision stimulus 20d to the behavior 10 and possibly adjust the parameters of these stimuli 20 using the stimuli/response editor 1140. In FIG. 11C, the user adds a Move Response 30c to the behavior 10 and selects the direction and duration of the movement using the stimuli/response editor 1140. FIG. 11D shows that the user has added the image acquisition response 30g and the Drone Response 30f, which enables the use to select take off or land using the stimuli/response editor 1140. And FIG. 11E shows a Save button 1150 that enables the user to name and save the behavior 10, which can then be used in a brain 40, exported, exchanged, and posted on a cloud server store.

[0102] Viewing Real-Time Robot Sensor Data and Operating the Robot

[0103] FIGS. 12A–12D illustrate an interface 1200 that provides the "Control" component of the GUI. This interface 1200 shows real-time image data acquired by an image sensor on a real or virtual robot connected to the system. The interface 1200/Control component may be used to operate a robot in real-time, e.g., by enabling the user to manually operate/drive the robot and engage complex functions from the drive screen (e.g., activate a brain, swap brains). The user can reach this interface 1200 by selecting the "see robot view" button 706 on the navigation page 700 of FIG. 7 or the select robot screen 800 of FIG. 8B.

[0104] In general, the interface 1200 may enable use of a dial format and/or swipe mode on a single screen. For instance, dials may provide indications of possible robot actions and/or easily recognizable symbols or icons (e.g., in addition to or instead of text). The user interface may give the user the ability to playback a behavior via button press, to show and/or hide a heads-up display (HUD), and/or to customize a HUD. In some implementations, supported controls may include but are not limited to: two-dial control; swipe control; two-dial control and swipe control on the same screen; tilt control (e.g., using the iPad sensors, move the robot in the direction of a device tilt); and voice commands. For swipe control, the robot may move in the direction of the swipe and may continue moving until the user lifts his or her swiping finger. The interface may enable the user to create a pattern, by swiping, for the robot to follow. (In some implementations the interface may show a trail on the screen in the direction of the swipe.) Similarly, vertical flying control altitude may utilize two finger gestures. Similarly, voice commands may encompass a plurality of actions. Other commands may include: device-type commands (e.g., forward, stop, right, left, faster), pet-related commands (e.g., come, heel), and

NVIDIA Ex1002 Page 420 of 1081

other commands (e.g., wag, to move the iPhone in a Romotive Romo back and forth or to roll an Orbotix Sphero back and forth).

[0105] FIG. 12A shows display and control icons superimposed on image data from the robot connected to the system. In this example, a dial 1 enables the user to drive the robot. Other buttons in the interface enable actions such as take pictures 2 and videos 3. Elements of the display can show robot-specific telemetry information 4, Wi-Fi connectivity 5, battery level 6, help systems 7, and settings 8. Other items in the interface include a button 9 for adding or accessing robot-specific actions in the interface and buttons 10, 11, and 12 for causing the robot to move in certain directions. Additionally, a "brain" button 13 enables the user to turn on/off a specific brain that has been added to the robot or load a new brain.

[0106] FIG. 12B illustrates how to select a particular object for the robot to track or follow. The user engages a "follow" brain by pressing hand 1201 and selects an area 1210 surrounding the object, e.g., with a mouse or cursor or by pinching a touchscreen as illustrated by hands 1202a and 1202b. The user then selects a follow button 1212 as indicated by hand 1203. Once the user has selected the object and selected the follow button, the hardware-agnostic brain using an AI/ANN/DNN to recognize the object in image data from an image sensor on or in the robot. The hardware-agnostic brain issues commands to its motors that cause the robot to move in the direction of the object, e.g., by rolling, walking, hopping, or flying as appropriate given the robot's movement capabilities and the object's motion.

[0107] FIG. 12C depicts selection of a person while performing a sport stunt. The user selects the person to follow, and the response consists of keeping the image centered around the selected person by appropriately translating the shift of the bounding box into robot-specific motor controls. In FIG. 12D, a different brain is selected, whose purpose is to keep the center of the robot "gaze" around the object, either enabling the robot to "circumnavigate" the object, or centering the camera on the object while the operator maneuvers the robot (e.g., a sort of "autofocus" aid for the user). An "emergency stop" button 1205 allows the user to immediately stop all action on a mobile robot, such as a UAV, UUV, walking robot, or rolling robot. For a UAV, the emergency stop button 1205 may turn off the UAV's motors, causing the UAV to drop from the sky as a safety precaution.

[0108] In FIG. 12D, the brain comprises one behavior, with one stimulus being "when the selected object is in view" triggering the response "drive around the object for 360 degrees". In another behavior, called "center behavior while driving", there may be two stimuli: a first stimulus that is "when the selected object is in view" and a second stimulus that is "when the operator drives the robot." The response to these stimuli may be "modify drive command from

NVIDIA Ex1002 Page 421 of 1081

the user so that the object is still centered in the image." This could occur, for example, if the commands a drone to translate on the left, and the "center behavior while driving" is engaged, the left translation command is modified so that the drone translates to the left, but rotates to the right to keep the object in the center of focus.

[0109] Robot Knowledge Center

[0110] An exemplary user interface may provide a robot knowledge center that enables the user to label knowledge learned by the system, or by the collection of systems (e.g., a swarm of robots, or sensor network) connected to the user interface. Knowledge can include visual, auditory, or multimodal objects, locations in a map, the identity of a scene (or a collection of views of that scene), and higher-order knowledge extracted by more elementary one (e.g., conceptual knowledge derived by reasoning on sensor data). Example of higher, more complex knowledge can be derived by machine vision (e.g., OpenCV), AI/ANN/DNN algorithms that extract concept out of collections of simpler objects (e.g., heavy objects vs. light objects, animated objects vs. inanimate objects).

[0111] This robot knowledge center is accessible and editable in at least two ways. First, a user can access and/or edit the robot knowledge center during the process of building brains, e.g., in the process of using information from the robot knowledge center to define stimuli and/or responses. Second, while operating a robot with the GUI, a user can label new objects added to the robot knowledge center. Also, certain information from the knowledge center might be available in the Heads-Up Display (HUD) on the drive screen. For example, the HUD might show the map of the current room the robot is in, and a user could label the map via the interface.

[0112] FIG. 13 shows an example robot knowledge center. In this example, the knowledge of the robot is divided in people, objects, and places. In this particular example, people and objects views are populated automatically, e.g., by the sensory module 110 (people, objects) and navigation module 130 (places) in the system of FIG. 15A, which is described in greater detail below. The user can select, e.g., via a touch screen on a tablet or a mouse, a specific view of a person or object, and provide a verbal or iconic label. Additionally, the user can take multiple views of an object, person, location (map), or a scene (e.g., multiple views of the kitchen) and group them in a single entity that combines all those views (e.g., several views of "John", or a cup, or "John's kitchen"), e.g. via a drag/drop interface. The user can also edit the map generated by module 130 (places), by providing verbal and iconic (e.g., color) labels to specific areas of the environment mapped by the robot. These verbally or iconically defined objects, people, and places can be used by the stimulus/response system.

NVIDIA Ex1002 Page 422 of 1081

[0113] Generalized Application Program Interfaces (APIs) for Robot Control

[0114] In order to abstract the specific robot hardware away from the algorithms executed by the brain, the Software Development Kits (SDKs) and APIs provided by or acquired from robotics companies are wrapped into a generalized API as described below. In this generalized API, two different robots with a similar set of sensors and hardware configurations would have the same set of API calls. If the two robots are extremely different, such as a robot capable of flight and a robot incapable of flight, then a subset of algorithms may prevent the robot with the more restrictive hardware configuration from performing incompatible actions (e.g., flying). However, a robot capable of flight can still learn and execute the algorithms that are used for navigation in a 2D space. This is because algorithms that execute in 2D space can still be executed on a UAV by ignoring the vertical axis in 3D space.

[0115] FIG. 14 shows a process layout for constructing a generalized API 70 suitable for interfacing between a robot or robot-specific API and a hardware-agnostic robot brain. This generalized API 70 is constructed in four process layers. In some embodiments, a single block is taken from Layer 1 of the process layout of API 70 that represents choosing a specific robot 72. In some embodiments, one or more blocks can be taken from Layer 2, as this is the step that configures hardware capabilities 74 of the chosen robot 72. Layer 3 is determined by the robot's movement capabilities 76, such as for example whether the robot 72 is a ground based robot, a UAV or a UUV. The final process step for Layer 4 is added for all robots as general commands 78, regardless of the selections and/or combinations of the previous process layers.

[0116] The generalized API 70 shown in FIG. 14 can be implemented in any suitable computing device or combination of computing devices. For example, the control device for the generalized API can be a mobile device (e.g., an iOS/Android mobile device), a single board computer, or a laptop/desktop computer. When the first instance of a particular robot is added to the system, the developer may check the robot manufacturer's API and hook the robot manufacturer's API into the generalized API 70. The process of coupling a robot-specific API from a robot manufacturer to the generalized 70 may be simpler/easier if the robot-specific API is similar to a previously configured robot-specific API.

[0117] The first layer checks for the specific robot 72 that is being connected. Based off of this information, the protocol that will be used to communicate with the robot 72 is determined as some robots use Bluetooth, some use the User Datagram Protocol (UDP), some use the Transport Control Protocol (TCP), etc. This also determines how the robot 72 connects to the system. Finally, this step determines if this robot has any robot-specific commands cannot be generalized to other robotic platforms. For example, a Jumping Sumo has a couple of jumping

NVIDIA Ex1002 Page 423 of 1081

options. For specific commands like these, the system provides an interface to allow developers to use them for specific projects, but with one major caveat: a warning is triggered when these robot-specific commands are used in standard algorithms, since these algorithms are intentionally generic.

[0118] The next layer search for hardware capabilities 74 of the robot 72, such as for example the available sensors on the robot 72 and sets up an API for those. Certain sensors can be used in place of each other (for example, infrareds and ultrasonic will both detect an object immediately in front of them). The algorithm itself defines this property, as it can be difficult to generalize if sensors can substituted without knowing the context in which they will be used. To continue with the previous example, if ultrasonic and infrared are only outputting a binary result (e.g., they see something or if they don't see something), then they can be reasonably substituted. However, if the algorithm requires an exact distance value as an output and this distance value is out of range for other sensors, then the algorithm can prevent substitution of sensors.

[0119] The next layer adds movement capabilities 76 of the robot 72, such as for example the number of dimensions (e.g. degree of freedom) the robot 72 can perform. Robots that can traverse underwater, such as for example UUVs or robots that can fly through the air, such as for example UAVs can maneuver in three dimensions. Ground robots, such as for example walking or wheeled robots can perform one-dimensional or two-dimensional algorithms.

[0120] The final layer adds generic commands 78 that apply to any robotics platform. For example, this layer adds one or more functions for connecting to and disconnecting from the robot 72, turning the robot 72 on and off, checking the robot's power supply, obtaining status information from the robot 72, etc.

[0121] The library, which may be stored in a memory or database, that handles generalizing across robotic structures has to make specific effort to abstract away the heterogeneous communication protocols. Each of these communication protocols has their own set of inherent properties. For example, UDP is connectionless and tends to be unreliable while TCP is connection-based and tends to be reliable. To abstract away these differences while maintaining a single API for all robots, helper objects are provided in the library to add some of those properties to communication protocols that don't have them inherently. For example, there is a reliable UDP stream to allow us to use communication paradigms that require reliability. This allows us to treat heterogeneous communication protocols as functionally similar which provides more flexibility for what algorithms can be used on robots.

NVIDIA Ex1002 Page 424 of 1081

[0122] One advantage of this approach is that the processor(s) can run the algorithms if the minimum hardware requirements are met or if sensors can be reasonably substituted for each other. This allows use of generalized algorithms that can be written on cheaper platforms with fewer features but that also run on more advanced platforms. However, there also exists the case where a developer is trying to run an algorithm on a robot that does not have the hardware to support it. Consider, for example, a ground-based robot with no camera is given an algorithm that requires it to fly around and record a video. To handle this case, each algorithm may provide a minimum hardware requirement.

[0123] Integration with Autonomous Behaviors (Autonomy)

[0124] The brains (collection of behaviors) described herein can be combined and associated with other forms of autonomous behaviors, such as autonomous sensory object recognition (such as but not limited to audition, vision, radio signals, LIDAR, or other point-cloud input, as well as any combination of the above sensors), in at least the following ways.

[0125] FIG. 15A depicts one example application where a (real or virtual) robot 100 is controlled by at least two autonomy modules (a sensory object module 110 and a motivation module 120) and several user-defined behaviors 160. The robot implements the autonomous,user-defined behaviors using various on-board sensors (e.g., cameras, accelerometer, gyro, IR, etc) that form a robot sensory system 100 and actuators/effectors (e.g., motors in tracks/propellers). The robot may also be linked to other sensors on associated hardware (e.g., a cell phone mounted on the robot, or a controlling iPad) that provide sensory input to an artificial robotic brain executing machine vision (e.g., OpenCV), AI, ANN, and DNN algorithms. These algorithms may be executed by:

a) a sensory module 110, which can be implemented, for instance, as a DNN processing video/audio/radio input, as well other sensory streams available from the robot (e.g., LIDAR, or other point cloud systems);
b) a motivation module 120, which can be implemented, for instance, as a spiking ANN to provide behavioral drives to the robot (e.g., curiosity to explore, need to recharge the battery, fear of colliding with an object, etc);
c) a navigation module 130, which can be implemented, for instance, as a continous-firing ANN learning the spatial layout of the environment and the location of the robot and objects in it.

Module 100 also provides access to robot motors/effectors via a motor control module 150.

NVIDIA Ex1002 Page 425 of 1081

[0126] Additionally, the robotic brain may be configured with an arbitrary number of behaviors (e.g., pairs of stimulus/response sets 160). Behaviors can be created and edited by the user based on stimuli/responses defined above (e.g., stimuli directly based on reading and preprocessing of robot sensors). They can also be chosen from a collection of stimuli/responses directly generated by machine vision (e.g., OpenCV) AI/ANN/DNN algorithms in the sensory objects module 110 and navigation modules 130. For example, a particular behavior can be defined to include predetermined stimuli, such as time of the day (e.g., it's 2:00PM as determined by the Robot processor, or the controlling cell phone), or a stimulus learned by the sensory system 110 (e.g. what "John" looks like). Similarly, a response associated with the behavior and executed in response to the stimulus can be defined from the navigation system 130 as "go to the kitchen." The resulting behavior would cause the robot to go to the kitchen (as learned by the navigation system) when the robot sees John, as learned by elaborating video/audio sensory information and/or other signals (e.g., wireless signals originating from John's phone).

[0127] The robot may also include a scheduler 140 that regulates control of the motor control 150 by the autonomous system and the user-defined system. For instance, the scheduler 140 may issue commands to a given robotic effector (e.g., a motor) in a sequential fashion rather than all at once. Behaviors 160 take control of motor control 150 after interacting with the scheduler 140. Motor control 150 in turns can control robot effectors in 100.

[0128] In the example instantiation in FIG. 15A, at least two autonomy modules 110, 120 and several user-defined behaviors 160 can control robotic effectors via the scheduler 140. For example, the sensory module 110 could command the robot to make a camera movement to learn more about an object visual appearance with a right movement of the robot, the navigation system 130 may command the robot to explore the environment with a left movement of the robot, and the behavior 160 may command the robot to go backward following the appearance of a soccer ball. As an example instantiation, the scheduler can use a neural-like competitive cueing network (or ANN, or DNN) to appropriately sequence actions based on their relative importance and timing.

[0129] FIG. 15B is a flowchart that illustrates an example scheduling process executed by the scheduler 140. The scheduler 140 starts by sorting its inputs and then computing the relative weight of each input. Inputs can come from a variety of sources, including on-board sensors 100, off-board sensors, and user inputs, and the scheduler can scale from having one input to many. Generally speaking, the input sources include modules currently executing algorithms (e.g., the navigation module 130), the motivation of the robot (motivation module 120),

NVIDIA Ex1002 Page 426 of 1081

command packets coming from a controller, and the currently executing brain (behaviors 160). Inputs with the highest weight execute, while inputs with lower weights that do not conflict with other inputs execute if they pass through a series of checks.

[0130] Each source coming into the scheduler 140 has more than one associated weight that gets combined into a final weight used by the scheduler 140. Each packet received by the scheduler 140 may have a specific weight for its individual command and a global weight provided by the scheduler 140 for that specific input. For example, if the scheduler 140 receives two motor commands from a controller—a first motor command with a global system weight of 0.2 and a specific weight of 0.4 and a second motor command with an global system weight of 0.1 and a specific weight of 0.9—it executes the second motor command as the combined weight of the second motor command is greater than that of the first motor command.

[0131] Global weights are typically determined by application developers and take into consideration priorities on the application level. For instance, a user input command may have priority over an algorithmically generated command (e.g., a STOP command from the drive screen may override a drive command generated by an AI/ANN/DNN). Likewise, global weights may take into account resource availability on a particular device.

[0132] Specific weights may be determined by the user of the application during the creation of the brain through positioning the behavior within the brain editor as described above with respect to FIG. 4A, 4B, 5A, and 10A–10C. Within each behavior, the specific weights may be refined algorithmically based on possibility of concurrent uses of resources. Furthermore, these specific weights can be adjusted during runtime depending on actual resource availability at the time of scheduling.

[0133] Beyond the basic series of weights, the scheduler 140 also executes one or more sorting steps. The first step involves sorting commands that use discrete hardware resources from commands that affect things like settings and parameter adjustment (operation 854). Settings changes are parsed and checked for conflict (operation 856). If there are no conflicts, then all settings changes push (operation 858). If there are conflicts and there are weights that can be used to break the conflict, they are used. If everything is weighted identically and two settings conflict, than neither executes or a symmetry-breaking procedure may be applied (e.g., most-used behavior wins). Many of these settings packets can be executed simultaneously. Next, the packets that affect discrete system resources are further sorted based on the affected resource(s) (operation 860). Commands that can inherently affect each other but don't necessarily do so are kept together. For example, audio playback and audio recording may be

NVIDIA Ex1002 Page 427 of 1081

kept in the same stream, as certain devices cannot record and playback and even if the option is available there are still constraints to deal with to avoid feedback.

[0134] For example, commands that affect motors may be grouped together. This allows decisions to be made while accounting for other packets that may conflict with the packet chosen to execute. In this particular implementation, If two packets have the potential to conflict but don't necessarily conflict, such as audio playback and audio recording, they may still be sent to the same group.

[0135] Once that packets that affect each individual hardware resource have been sorted into their own groups, the scheduler 140 determines which inputs to execute and the order in which to execute them (operation 862). The system checks if a resource is in use, and if it is, what was the weight of the packet that took control of the resource. In order to take control of a resource, a packet must have a higher weight than the packet that currently holds the resource. If it's a lower weight, it gets ignored and thrown away. If it's a higher weight, it takes control of the resource.

[0136] Different input sources can also communicate to each other and adjust the weights of the other subsystems. For instance, if the motivation system 120 is really interested in navigating, but it wants to navigate in a different direction, it can adjust the weights of the navigation packets being sent into the scheduler 140 by signaling the navigation system 130.

[0137] The scheduling process of FIG. 15B allows the robot to look for an object in the environment, then step backward as required by the user-defined behavior, then go on exploring the environment. In order to determine the relative importance of actions, the scheduler 140 may use the graphical placement of behaviors in the brain to determine the relative importance of each behavior in the brain. In other implementations, a user may be able to provide positive and/or negative reinforcement (e.g., during a training process with the robot) in order to train the robot to develop an understanding of which behaviors and/or responses to prioritize over others. In another implementation, an ANN/DNN autonomously prioritizes scheduling based on learning and experience. In another implementation, the user may manually define the importance of each behavior, e.g., determining which behavior gets the precedence over other behaviors when both behaviors comprise stimuli which would activate their two different sets of responses in reaction a single event. For example, when an image contains two stimuli (e.g., a face and the color red) which simultaneously activate two sets of responses, the user may manually pre-determine when behavior 1 is engaged and when behavior 2 may be performed if behavior 1 is not complete (e.g., the user may indicate that behavior 2 may interrupt behavior 1, may start after behavior 1 completes, and/or the like).

NVIDIA Ex1002 Page 428 of 1081

Conclusion

[0138] While various inventive embodiments have been described and illustrated herein, those of ordinary skill in the art will readily envision a variety of other means and/or structures for performing the function and/or obtaining the results and/or one or more of the advantages described herein, and each of such variations and/or modifications is deemed to be within the scope of the inventive embodiments described herein. More generally, those skilled in the art will readily appreciate that all parameters, dimensions, materials, and configurations described herein are meant to be exemplary and that the actual parameters, dimensions, materials, and/or configurations will depend upon the specific application or applications for which the inventive teachings is/are used. Those skilled in the art will recognize, or be able to ascertain using no more than routine experimentation, many equivalents to the specific inventive embodiments described herein. It is, therefore, to be understood that the foregoing embodiments are presented by way of example only and that, within the scope of the appended claims and equivalents thereto, inventive embodiments may be practiced otherwise than as specifically described and claimed. Inventive embodiments of the present disclosure are directed to each individual feature, system, article, material, kit, and/or method described herein. In addition, any combination of two or more such features, systems, articles, materials, kits, and/or methods, if such features, systems, articles, materials, kits, and/or methods are not mutually inconsistent, is included within the inventive scope of the present disclosure.

[0139] The above-described embodiments can be implemented in any of numerous ways. For example, embodiments of designing and making the technology disclosed herein may be implemented using hardware, software or a combination thereof. When implemented in software, the software code can be executed on any suitable processor or collection of processors, whether provided in a single computer or distributed among multiple computers.

[0140] Further, it should be appreciated that a computer may be embodied in any of a number of forms, such as a rack-mounted computer, a desktop computer, a laptop computer, or a tablet computer. Additionally, a computer may be embedded in a device not generally regarded as a computer but with suitable processing capabilities, including a Personal Digital Assistant (PDA), a smart phone or any other suitable portable or fixed electronic device.

[0141] Also, a computer may have one or more input and output devices. These devices can be used, among other things, to present a user interface. Examples of output devices that can be used to provide a user interface include printers or display screens for visual presentation of output and speakers or other sound generating devices for audible presentation of output. Examples of input devices that can be used for a user interface include keyboards, and pointing

NVIDIA Ex1002 Page 429 of 1081

devices, such as mice, touch pads, and digitizing tablets. As another example, a computer may receive input information through speech recognition or in other audible format. The computer may also receive input by visual observation (e.g., camera) or by a motion sensing device (e.g., Microsoft Knect).

[0142] Such computers may be interconnected by one or more networks in any suitable form, including a local area network or a wide area network, such as an enterprise network, and intelligent network (IN) or the Internet. Such networks may be based on any suitable technology and may operate according to any suitable protocol and may include wireless networks, wired networks or fiber optic networks.

[0143] The various methods or processes outlined herein may be coded as software that is executable on one or more processors that employ any one of a variety of operating systems or platforms. Additionally, such software may be written using any of a number of suitable programming languages and/or programming or scripting tools, and also may be compiled as executable machine language code or intermediate code that is executed on a framework or virtual machine.

[0144] In this respect, various inventive concepts may be embodied as a computer readable storage medium (or multiple computer readable storage media) (e.g., a computer memory, one or more floppy discs, compact discs, optical discs, magnetic tapes, flash memories, circuit configurations in Field Programmable Gate Arrays or other semiconductor devices, or other non-transitory medium or tangible computer storage medium) encoded with one or more programs that, when executed on one or more computers or other processors, perform methods that implement the various embodiments of the invention discussed above. The computer readable medium or media can be transportable, such that the program or programs stored thereon can be loaded onto one or more different computers or other processors to implement various aspects of the present invention as discussed above.

[0145] The terms "program" or "software" are used herein in a generic sense to refer to any type of computer code or set of computer-executable instructions that can be employed to program a computer or other processor to implement various aspects of embodiments as discussed above. Additionally, it should be appreciated that according to one aspect, one or more computer programs that when executed perform methods of the present invention need not reside on a single computer or processor, but may be distributed in a modular fashion amongst a number of different computers or processors to implement various aspects of the present invention.

NVIDIA Ex1002 Page 430 of 1081

[0146] Computer-executable instructions may be in many forms, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically the functionality of the program modules may be combined or distributed as desired in various embodiments.

[0147] Also, data structures may be stored in computer-readable media in any suitable form. For simplicity of illustration, data structures may be shown to have fields that are related through location in the data structure. Such relationships may likewise be achieved by assigning storage for the fields with locations in a computer-readable medium that convey relationship between the fields. However, any suitable mechanism may be used to establish a relationship between information in fields of a data structure, including through the use of pointers, tags or other mechanisms that establish relationship between data elements.

[0148] Also, various inventive concepts may be embodied as one or more methods, of which an example has been provided. The acts performed as part of the method may be ordered in any suitable way. Accordingly, embodiments may be constructed in which acts are performed in an order different than illustrated, which may include performing some acts simultaneously, even though shown as sequential acts in illustrative embodiments.

[0149] All definitions, as defined and used herein, should be understood to control over dictionary definitions, definitions in documents incorporated by reference, and/or ordinary meanings of the defined terms.

[0150] The indefinite articles "a" and "an," as used herein in the specification and in the claims, unless clearly indicated to the contrary, should be understood to mean "at least one."

[0151] The phrase "and/or," as used herein in the specification and in the claims, should be understood to mean "either or both" of the elements so conjoined, i.e., elements that are conjunctively present in some cases and disjunctively present in other cases. Multiple elements listed with "and/or" should be construed in the same fashion, i.e., "one or more" of the elements so conjoined. Other elements may optionally be present other than the elements specifically identified by the "and/or" clause, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, a reference to "A and/or B", when used in conjunction with open-ended language such as "comprising" can refer, in one embodiment, to A only (optionally including elements other than B); in another embodiment, to B only (optionally including elements other than A); in yet another embodiment, to both A and B (optionally including other elements); etc.

NVIDIA Ex1002 Page 431 of 1081

[0152] As used herein in the specification and in the claims, "or" should be understood to have the same meaning as "and/or" as defined above. For example, when separating items in a list, "or" or "and/or" shall be interpreted as being inclusive, i.e., the inclusion of at least one, but also including more than one, of a number or list of elements, and, optionally, additional unlisted items. Only terms clearly indicated to the contrary, such as "only one of" or "exactly one of," or, when used in the claims, "consisting of," will refer to the inclusion of exactly one element of a number or list of elements. In general, the term "or" as used herein shall only be interpreted as indicating exclusive alternatives (i.e. "one or the other but not both") when preceded by terms of exclusivity, such as "either," "one of," "only one of," or "exactly one of." "Consisting essentially of," when used in the claims, shall have its ordinary meaning as used in the field of patent law.

[0153] As used herein in the specification and in the claims, the phrase "at least one," in reference to a list of one or more elements, should be understood to mean at least one element selected from any one or more of the elements in the list of elements, but not necessarily including at least one of each and every element specifically listed within the list of elements and not excluding any combinations of elements in the list of elements. This definition also allows that elements may optionally be present other than the elements specifically identified within the list of elements to which the phrase "at least one" refers, whether related or unrelated to those elements specifically identified. Thus, as a non-limiting example, "at least one of A and B" (or, equivalently, "at least one of A or B," or, equivalently "at least one of A and/or B") can refer, in one embodiment, to at least one, optionally including more than one, A, with no B present (and optionally including elements other than B); in another embodiment, to at least one, optionally including more than one, B, with no A present (and optionally including elements other than A); in yet another embodiment, to at least one, optionally including more than one, A, and at least one, optionally including more than one, B (and optionally including other elements); etc.

[0154] In the claims, as well as in the specification above, all transitional phrases such as "comprising," "including," "carrying," "having," "containing," "involving," "holding," "composed of," and the like are to be understood to be open-ended, i.e., to mean including but not limited to. Only the transitional phrases "consisting of" and "consisting essentially of" shall be closed or semi-closed transitional phrases, respectively, as set forth in the United States Patent Office Manual of Patent Examining Procedures, Section 2111.03.

NVIDIA Ex1002 Page 432 of 1081

Claims

1. A method for generating a hardware-agnostic behavior of at least one electronic device, the method comprising:

A) receiving, via a user interface, at least one stimulus selection from a user, the at least one stimulus selection corresponding to at least one stimulus detectable by the at least one electronic device;

B) receiving, via the user interface, at least one hardware-agnostic response selection from the user, the at least one hardware-agnostic response selection corresponding to at least one action to be performed by the at least one electronic device in response to the at least one stimulus; and

C) generating, via a processor operably coupled to the user interface, the hardware-agnostic behavior based on the at least one stimulus selection received in A) and the at least one hardware-agnostic response selection received in B).

2. The method of claim 1, wherein the at least one electronic device comprises at least one robot.

3. The method of claim 1, wherein the at least one stimulus is based at least in part on an output from a neural network.

4. The method of claim 1, wherein the at least one stimulus comprises sensing at least one of:

depressing a button;
swiping a touchscreen;
a change in attitude with a gyroscope;
acceleration with an accelerometer;
a change in battery charge;
a wireless signal strength;
a time of day;
a date;
passage of a predetermined time period;
magnetic field strength;
electric field strength;
stress;
strain;
position;

NVIDIA Ex1002 Page 433 of 1081

altitude;

speed;

velocity;

angular velocity;

trajectory;

a face, object, and/or scene with an imaging detector;

motion;

touch; and

sound and/or speech with a microphone.

5. The method of claim 1, wherein the at least one response is based at least in part on an output from a neural network.

6. The method of claim 5, wherein the output from the neural network comprises at least one of a visual object or an auditory object recognized by the neural network.

7. The method of claim 1, further comprising:

D) receiving, via the user interface, a selection of at least one particular electronic device to associate with the hardware-agnostic behavior generated in C); and

E) associating the hardware-agnostic behavior generated in C) with the at least one particular electronic device selected in D).

8. The method of claim 7, wherein E) comprises:

determining identifying information for the at least one particular electronic device, the identifying information including information about at least one sensor and/or at least one actuator associated with the at least one particular electronic device.

9. The method of claim 8, wherein E) comprises:

translating the hardware-agnostic behavior into hardware-specific instructions based at least in part on the identifying information; and

providing the hardware-specific instructions to the at least one particular electronic device.

10. The method of claim 1, wherein the at least one hardware-agnostic response selection comprises a sequence of actions to be performed by the at least one electronic device in response to at least one corresponding stimulus.

NVIDIA Ex1002 Page 434 of 1081

11. The method of claim 1, further comprising:

F) generating, via the processor, at least one other hardware-agnostic behavior based on at least one other stimulus selection and at least one other hardware-agnostic response selection; and

G) forming a hardware-agnostic personality based at least on the hardware-agnostic robot behavior and at least one other hardware-agnostic robot behavior.

12. A system for generating a hardware-agnostic behavior of at least one electronic device, the system comprising:

a user interface to receive:

at least one stimulus selection from a user, the at least one stimulus selection corresponding to at least one stimulus detectable by the at least one electronic device; and

at least one hardware-agnostic response selection from the user, the at least one hardware-agnostic response selection corresponding to at least one action to be performed by the at least one electronic device in response to the at least one stimulus;

a processor, operably coupled to the user interface, to generate the hardware-agnostic behavior based on the at least one stimulus selection and the at least one hardware-agnostic response selection; and

a communications port, operably coupled to the processor, to provide the hardware-agnostic behavior to the at least one electronic device.

13. The system of claim 12, further comprising:

a hardware translation component, operably coupled to the communications port, to translate the hardware-agnostic behavior into a set of hardware-specific input triggers to be sensed by the at least one electronic device and a set of hardware-specific actions in response to the set of hardware-specific input triggers to be performed by the at least one electronic device.

14. A computer-implemented method for loading at least one hardware-agnostic behavior between a first robot and a second robot, the method comprising:

receiving a request to load a first hardware-agnostic behavior onto the first robot;

retrieving the first hardware-agnostic behavior from at least one storage device, the first hardware-agnostic behavior defining at least one first hardware-agnostic robot response to at least one first hardware-agnostic robot sensor stimulus;

providing the first hardware-agnostic behavior to the first robot;

providing the first hardware-agnostic behavior to the second robot;

NVIDIA Ex1002 Page 435 of 1081

receiving a request to load a second hardware-agnostic behavior onto the first robot, the second hardware-agnostic behavior defining at least one second hardware-agnostic robot response to at least one second hardware-agnostic robot sensor stimulus;

retrieving the second hardware-agnostic behavior from the at least one storage device; and

providing the second hardware-agnostic behavior to the first robot.

15. The method of claim 14, further comprising:

providing the second hardware-agnostic behavior to the second robot.

16. The method of claim 14, wherein providing the second hardware-agnostic behavior to the first robot comprises replacing, in the robot, the first hardware-agnostic behavior with the second hardware-agnostic behavior.

17. A computer-implemented method for generating behaviors for a robot, the method comprising:

receiving, at a user interface, a selection of at least one stimulus to be sensed by the robot;

receiving, at the user interface, a selection of at least one response to be performed by the robot;

generating a behavior for the robot based at least in part on the at least one stimulus and the at least one response; and

rendering, via the user interface, the behavior as a behavior neuron comprising at least one dendrite representing the at least one stimulus and at least a portion of a neuron axon representing the at least one response.

18. The method of claim 17, wherein rendering the behavior comprises:

rendering the behavior neuron as at least one neuron of a plurality of neurons in a graphical representation of a brain.

19. The method of claim 18, wherein rendering the behavior neuron comprises:

rendering the at least one neuron in the graphical representation of the brain based on the nature of the behavior in relation to behavior centers of an animal brain.

NVIDIA Ex1002 Page 436 of 1081

20. A method of engaging at least one hardware-agnostic behavior to control at least one robot, the hardware-agnostic behavior comprising at least one action to be performed by the at least one robot in response to the at least one stimulus sensed by the at least one robot, the method comprising:

establishing a communications connection between the at least one robot and a graphical user interface (GUI);

receiving, via the GUI, an indication from a user regarding selection of the at least one hardware-agnostic behavior;

retrieving, from a memory operably coupled to the GUI, instructions for causing the at least one robot to operate according to the at least one hardware-agnostic behavior ; and

executing the instructions with a processor operably coupled to the GUI so as to engage the at least one hardware-agnostic behavior to control the at least one robot.

NVIDIA Ex1002 Page 437 of 1081

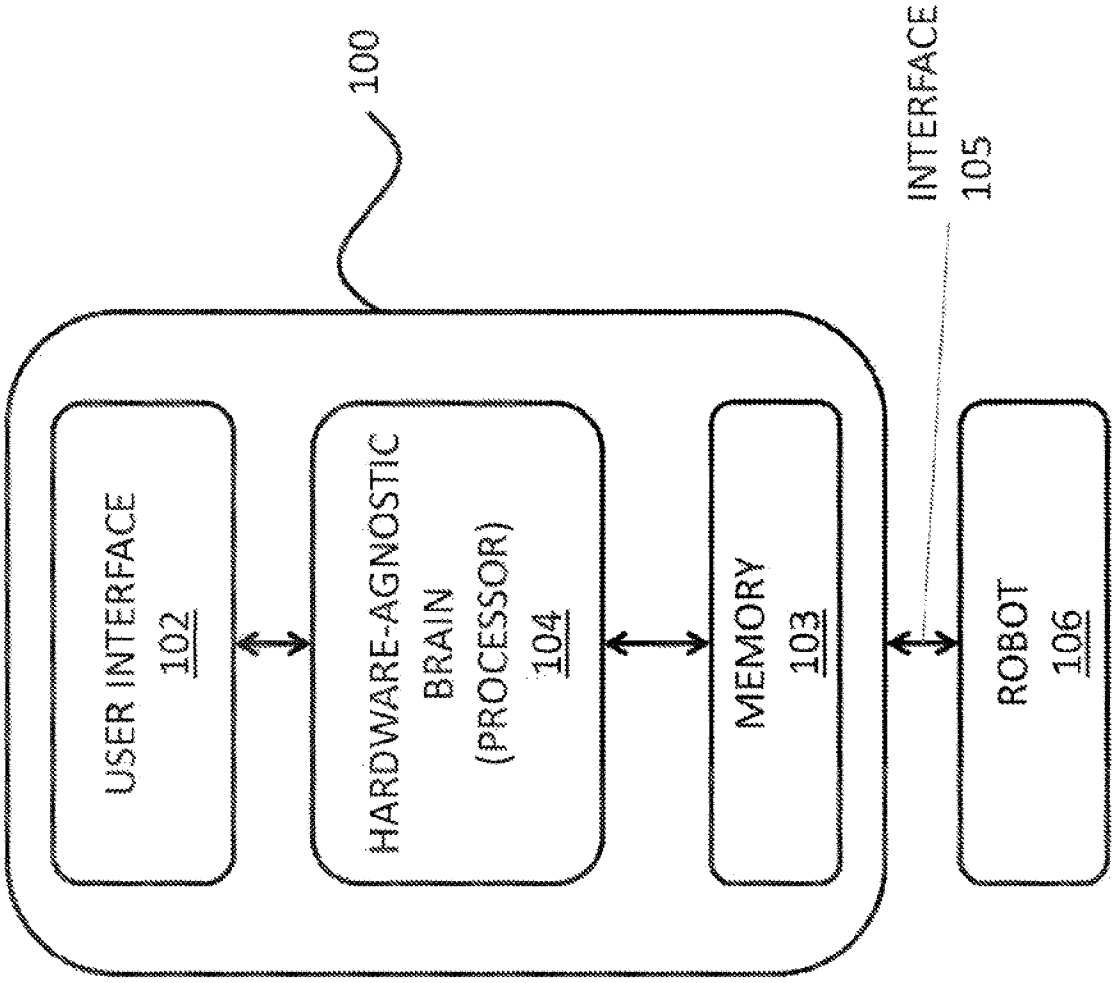

100
INTERFACE
105
USER INTERFACE
102
HARDWARE-AGNOSTIC
BRAIN
(PROCESSOR)
104
MEMORY
103
ROBOT
106

Fig. 1

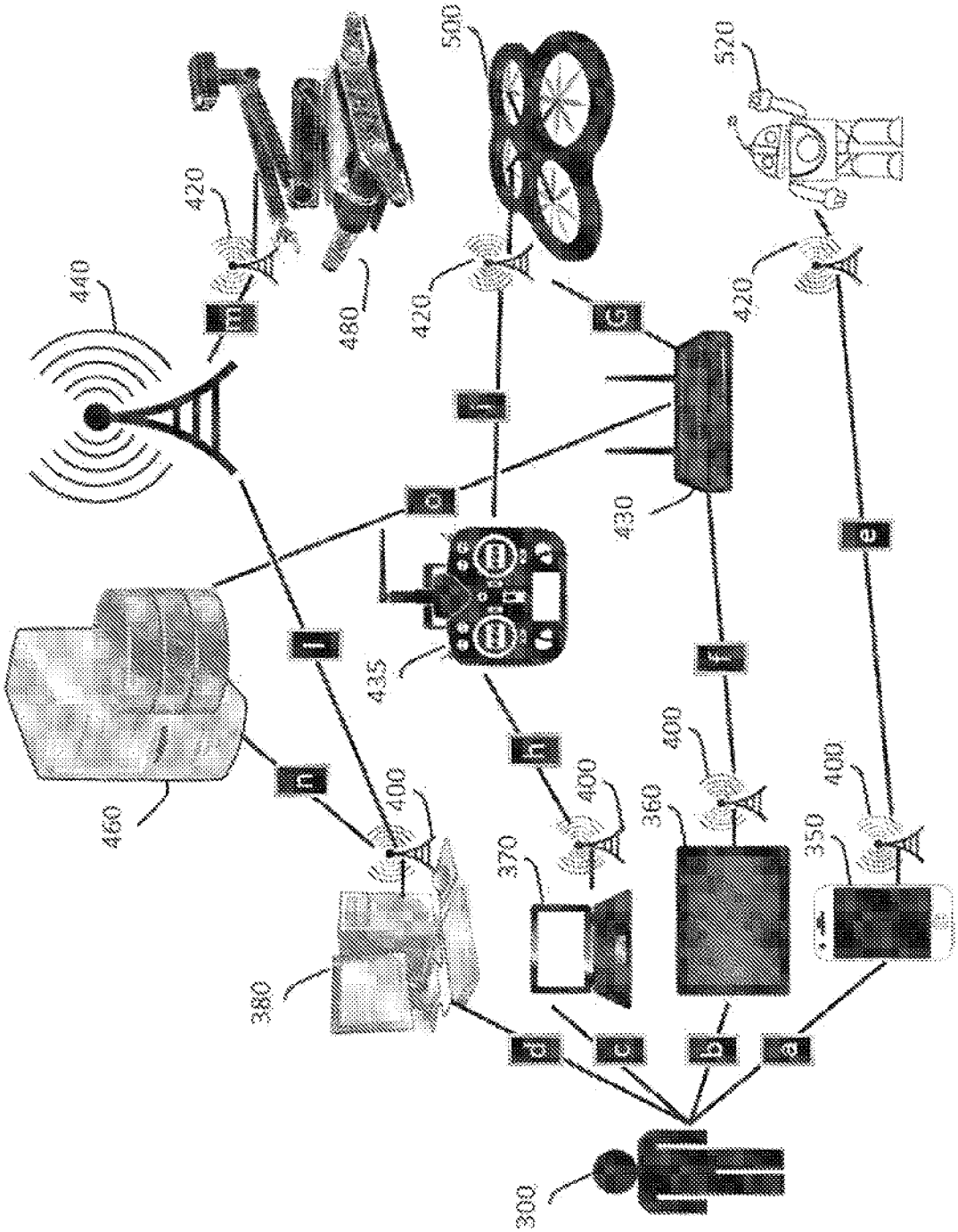

300
310
320
330
340
350
360
370
380
390
400
410
420
430
435
440
450
460
470
480
490
500
510
520

Fig. 2

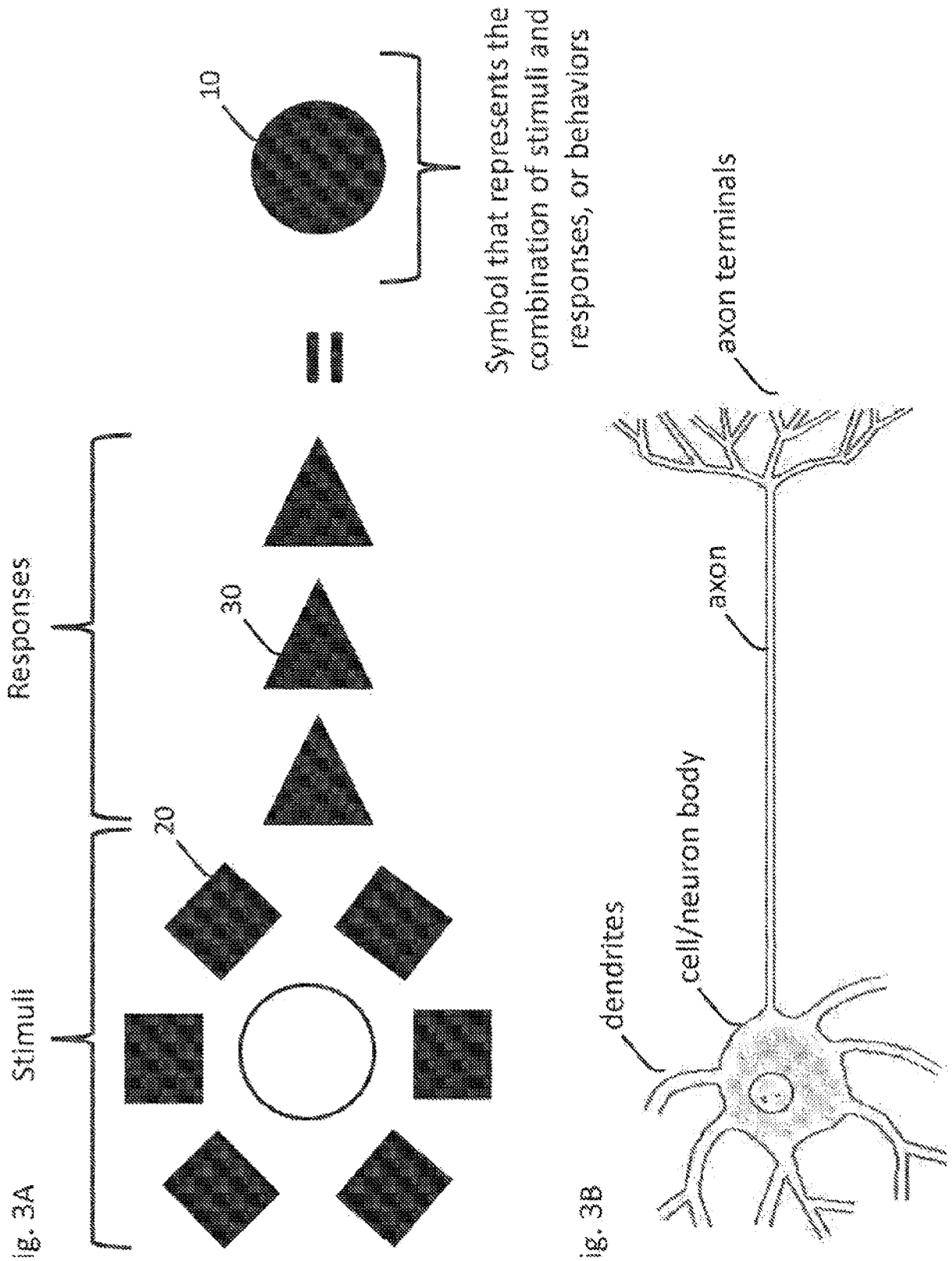

Stimuli
Responses
20
30
10
Symbol that represents the combination of stimuli and responses, or behaviors

Fig. 3A

dendrites
cell/neuron body
axon
axon terminals

Fig. 3B

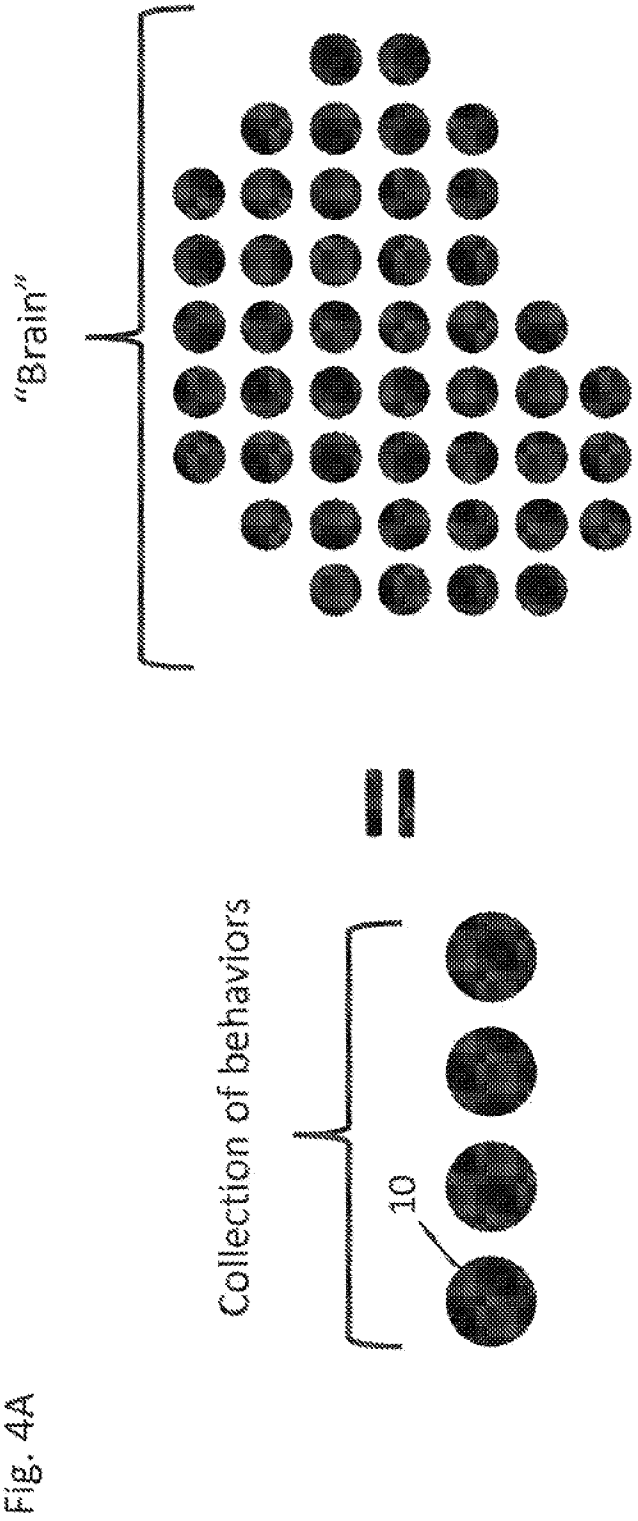


Fig. 4A

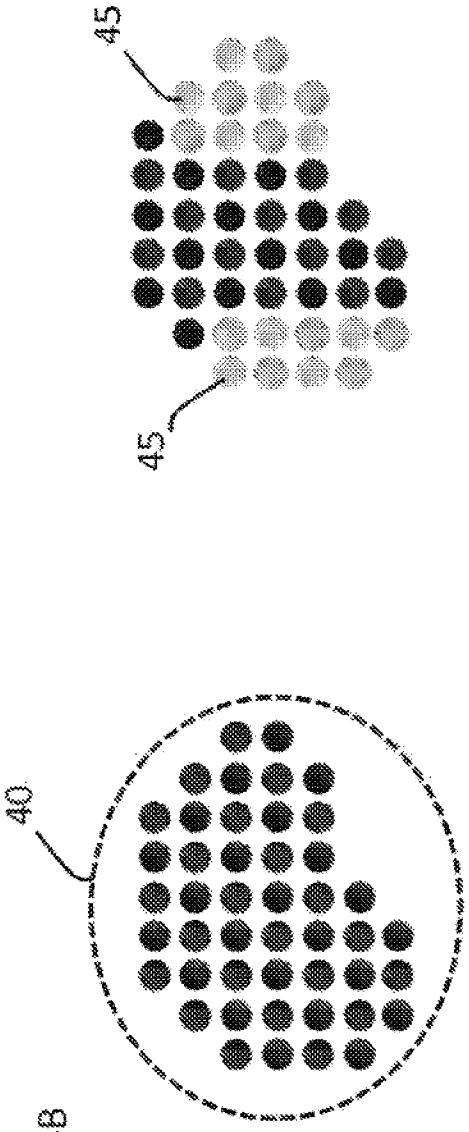


Fig. 4B

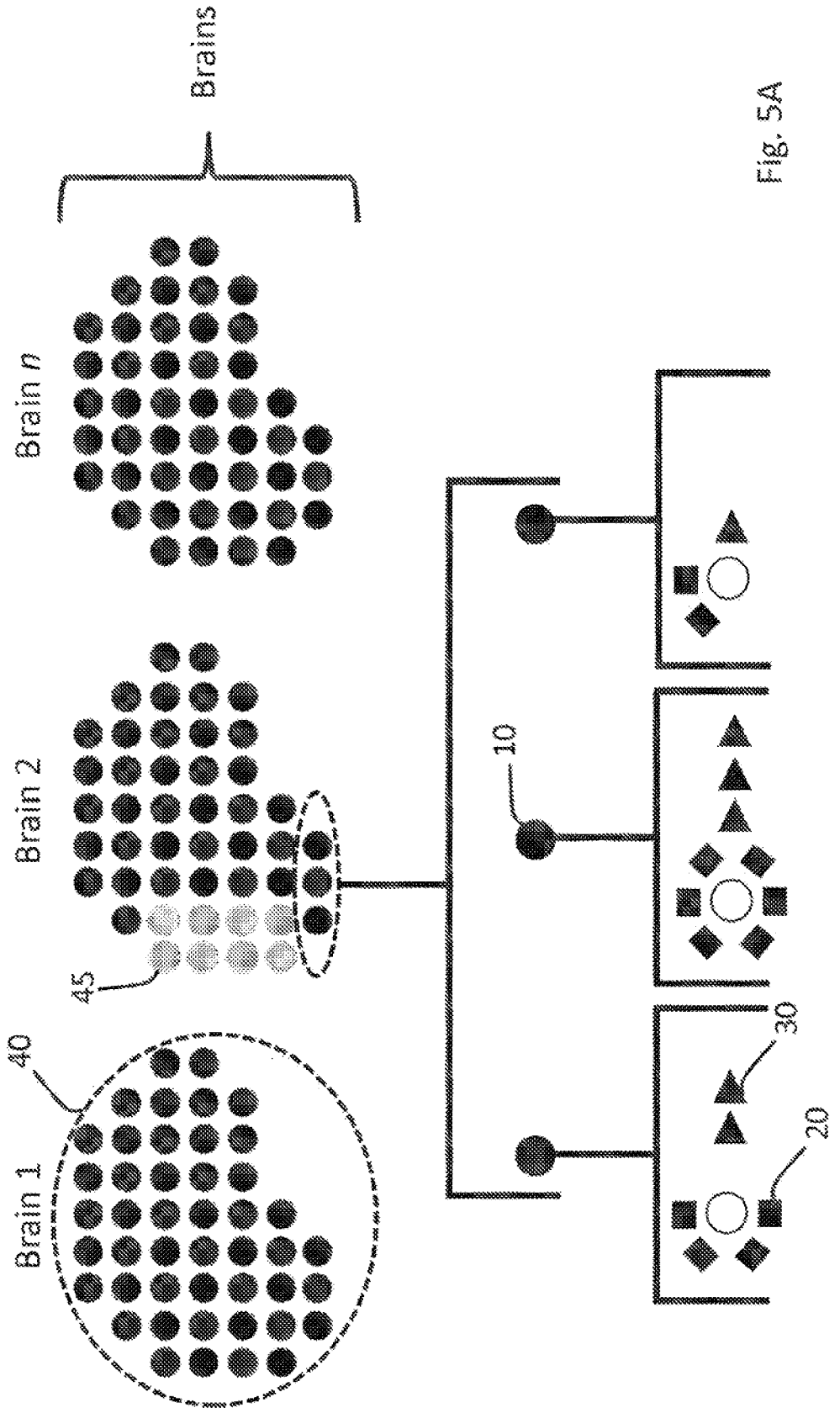
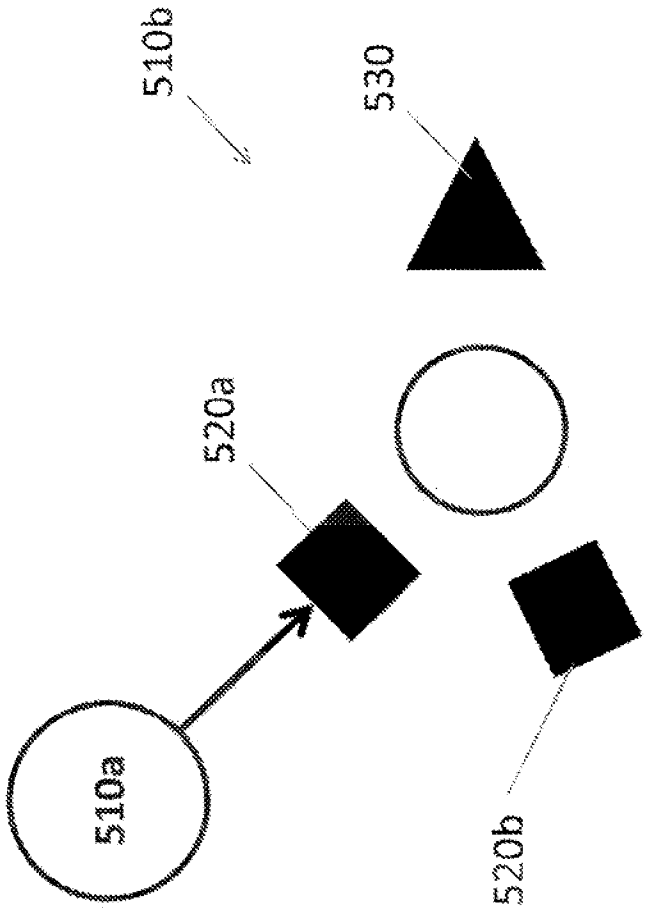


Fig. 5A

Fig. 5B



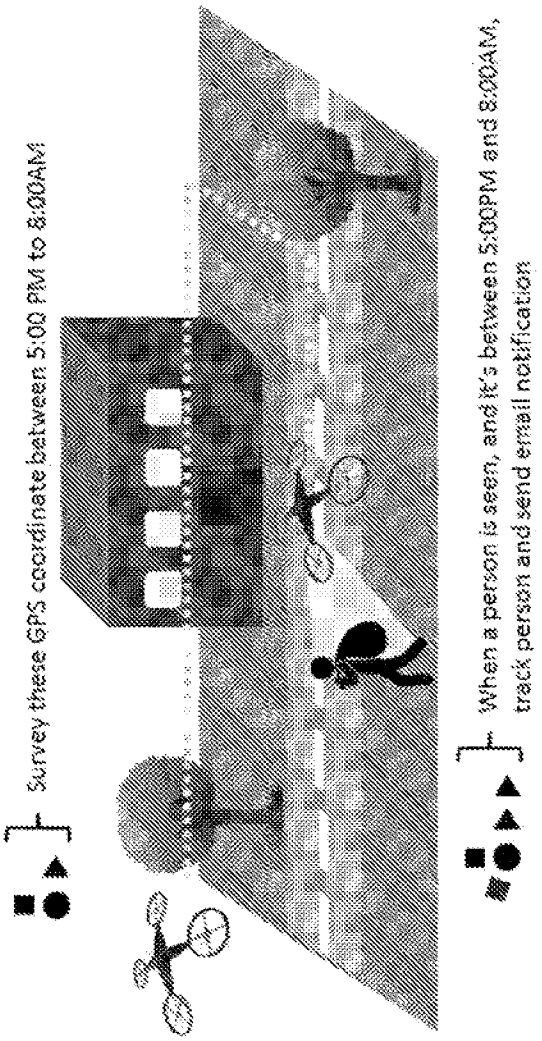

Survey these GPS coordinate between 5:00 PM to 8:00AM
When a person is seen, and it's between 5:00PM and 8:00AM, track person and send email notification

Fig. 6

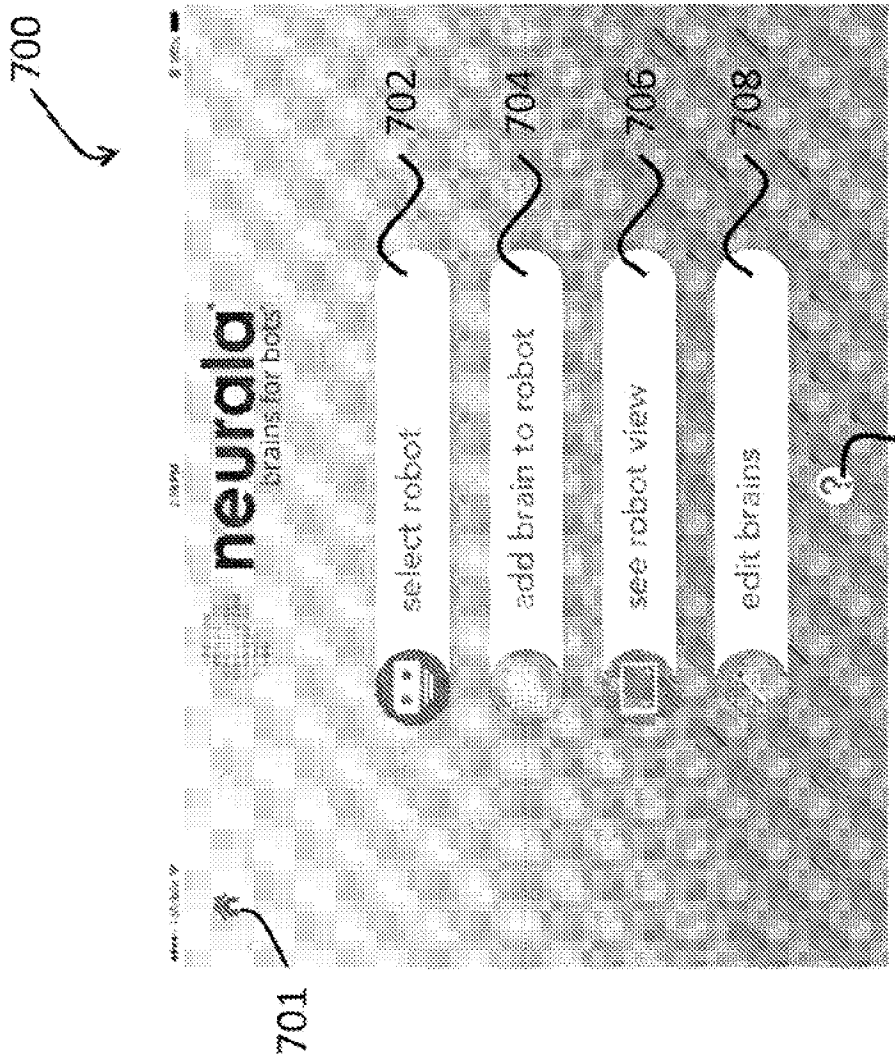

700
701
neurala
brains for bots
select robot
add brain to robot
see robot view
edit brains
702
704
706
708
710

Fig. 7

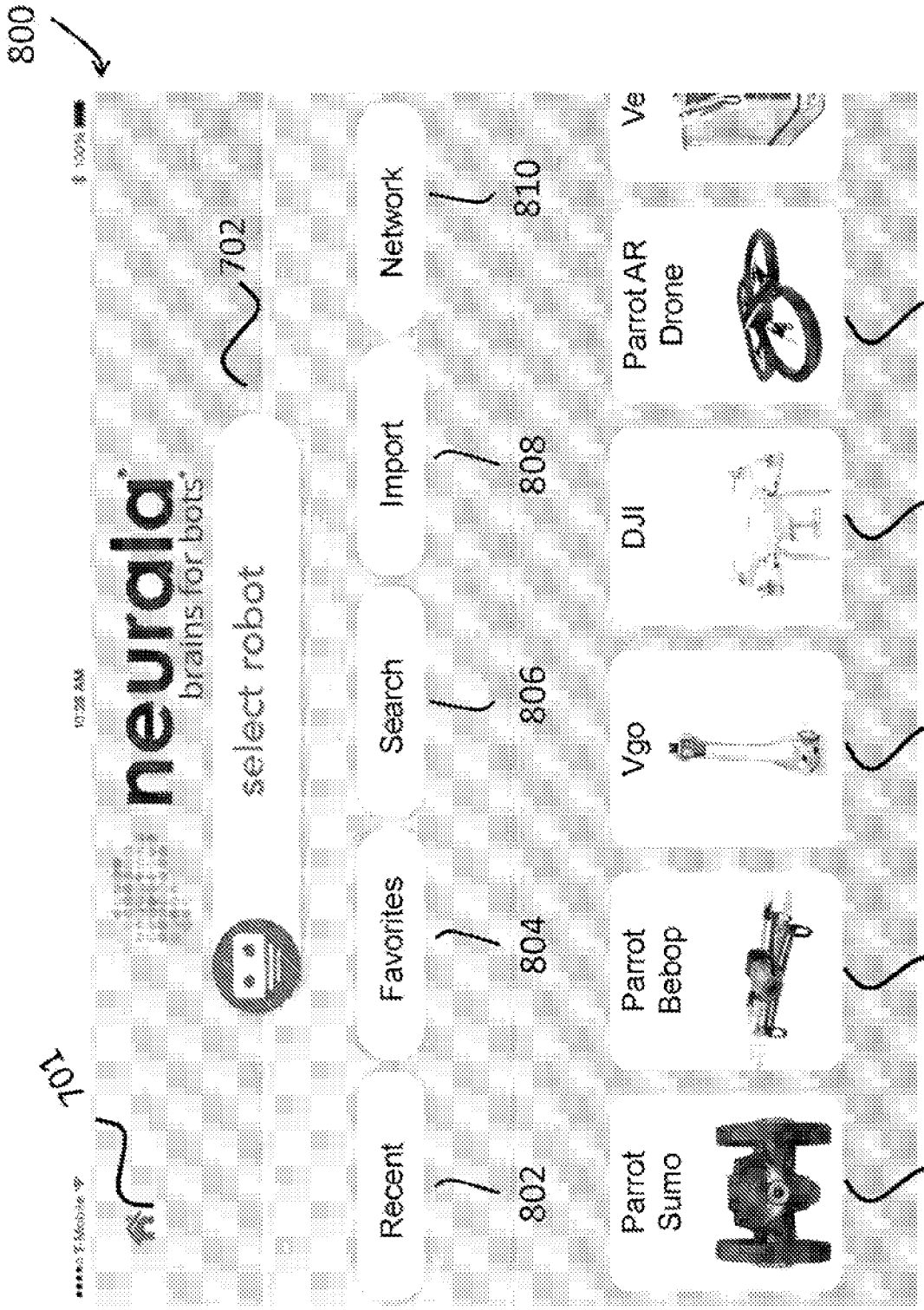

800
neurala
brains for bots
select robot
Recent
Favorites
Search
Import
Network
802
804
806
808
810
Parrot Sumo
Parrot Bebop
Vgo
DJI
Parrot AR Drone
820a
820b
820c
820d
820e
702

Fig. 8A

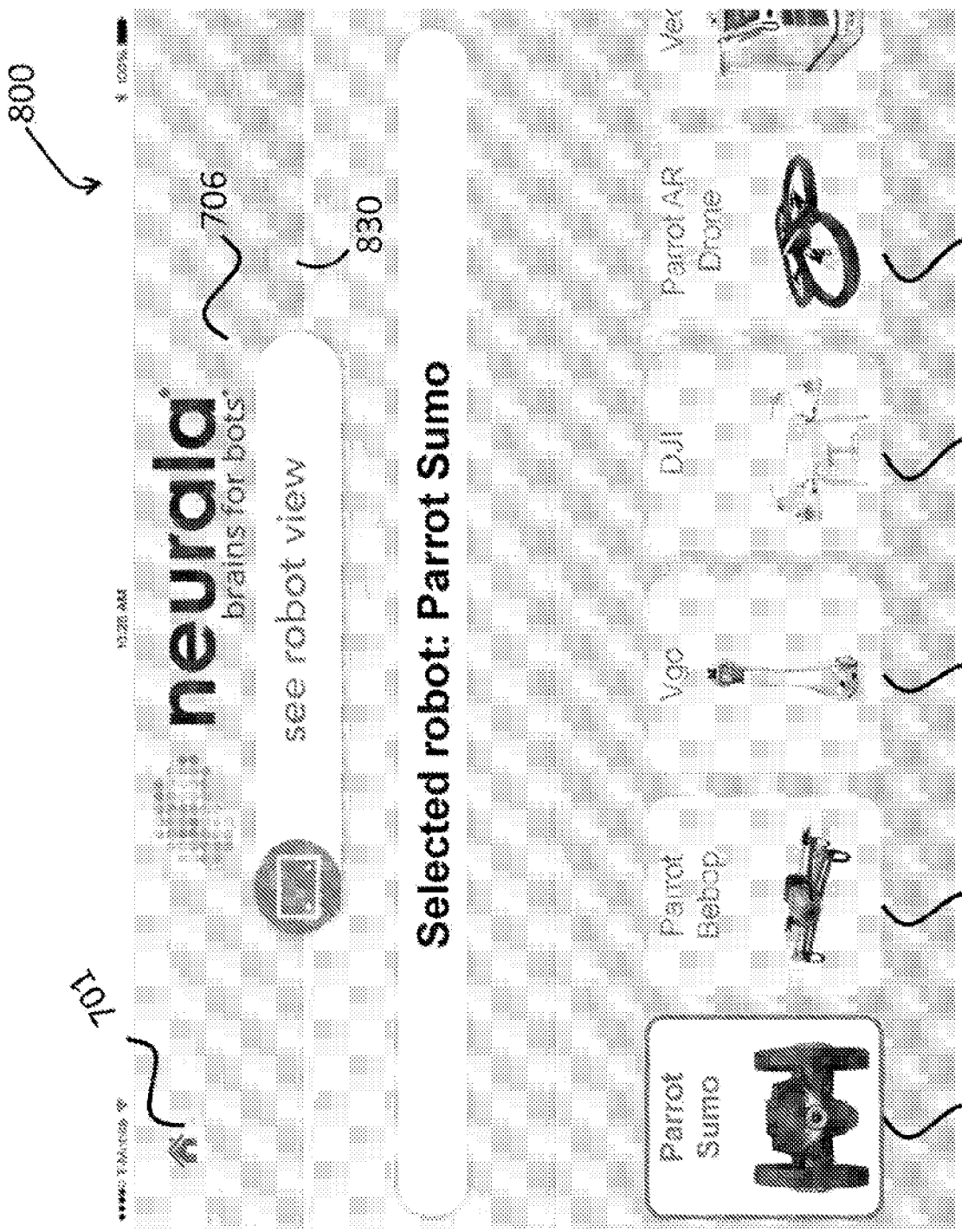

800
702
706
830
neurala
brains for bots
see robot view
Selected robot: Parrot Sumo
Parrot Sumo
Parrot Bebop
Vgo
DJI
Parrot AR Drone
820a
820b
820c
820d
820e

Fig. 8B

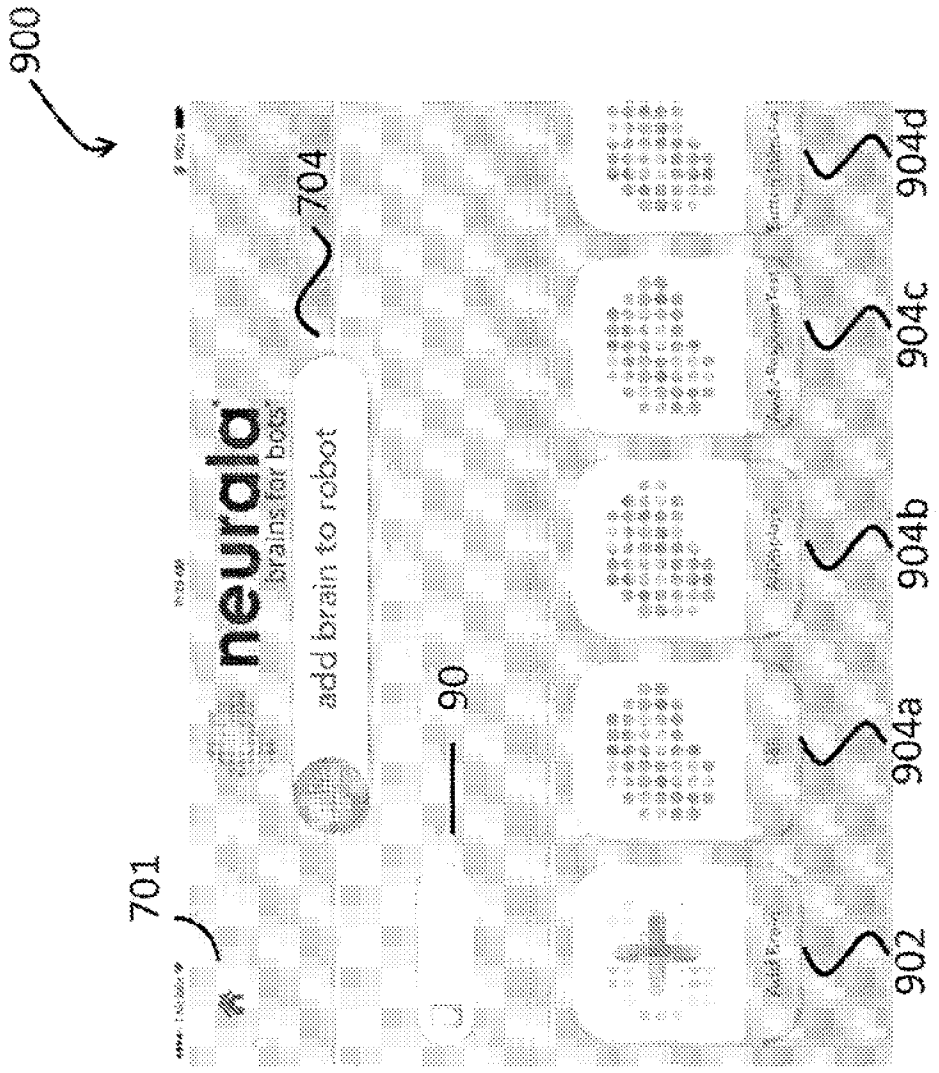

900
701
neurala
brains for bots
704
add brain to robot
90
902
904a
904b
904c
904d

Fig. 9A

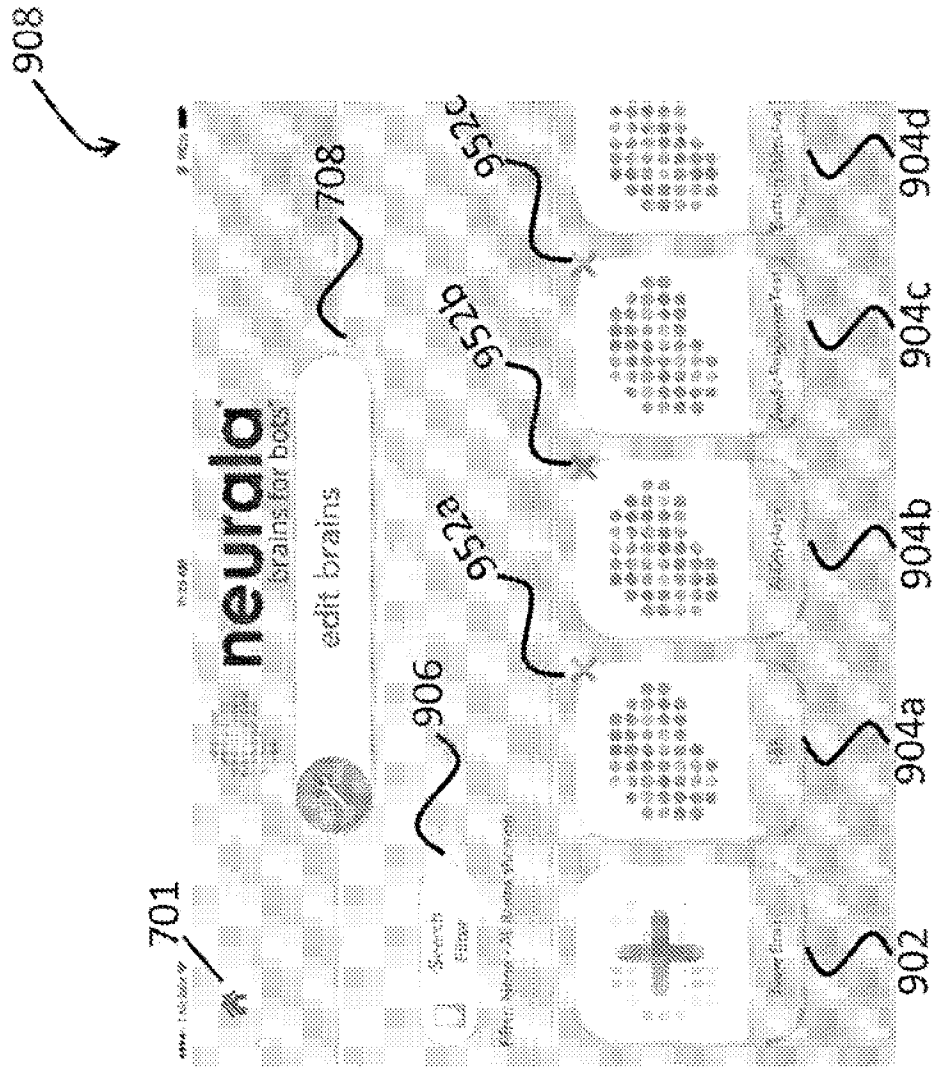

908
701
neurala
brains for bots
edit brains
708
906
952a
952b
952c
902
904a
904b
904c
904d

Fig. 9B

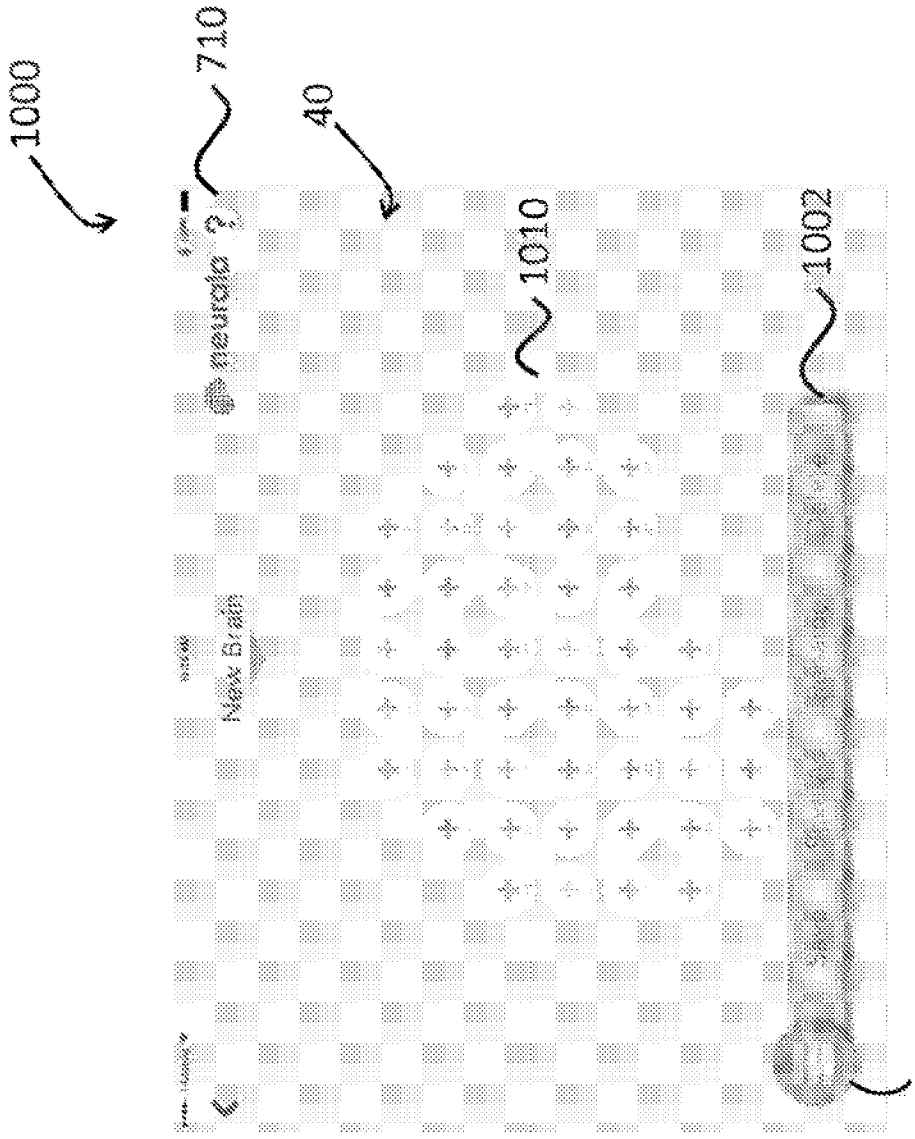


Fig. 10A

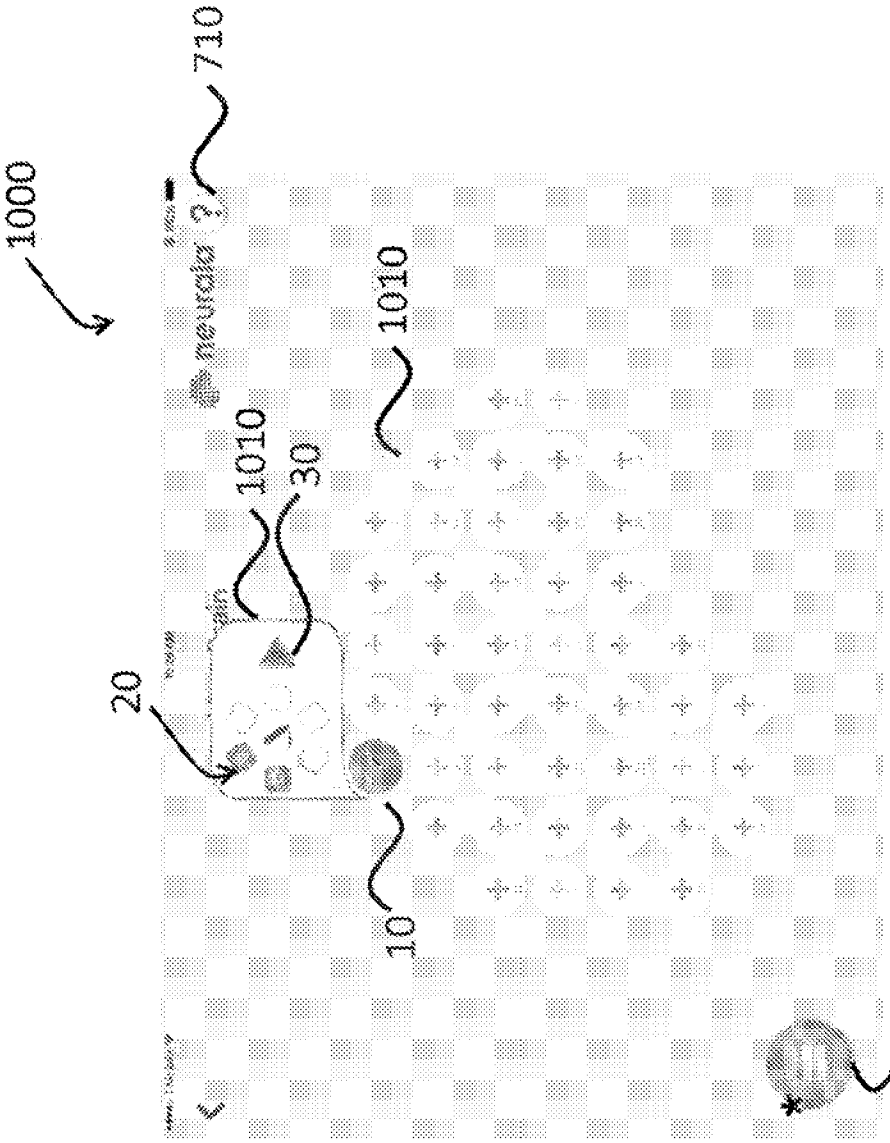


Fig. 10B

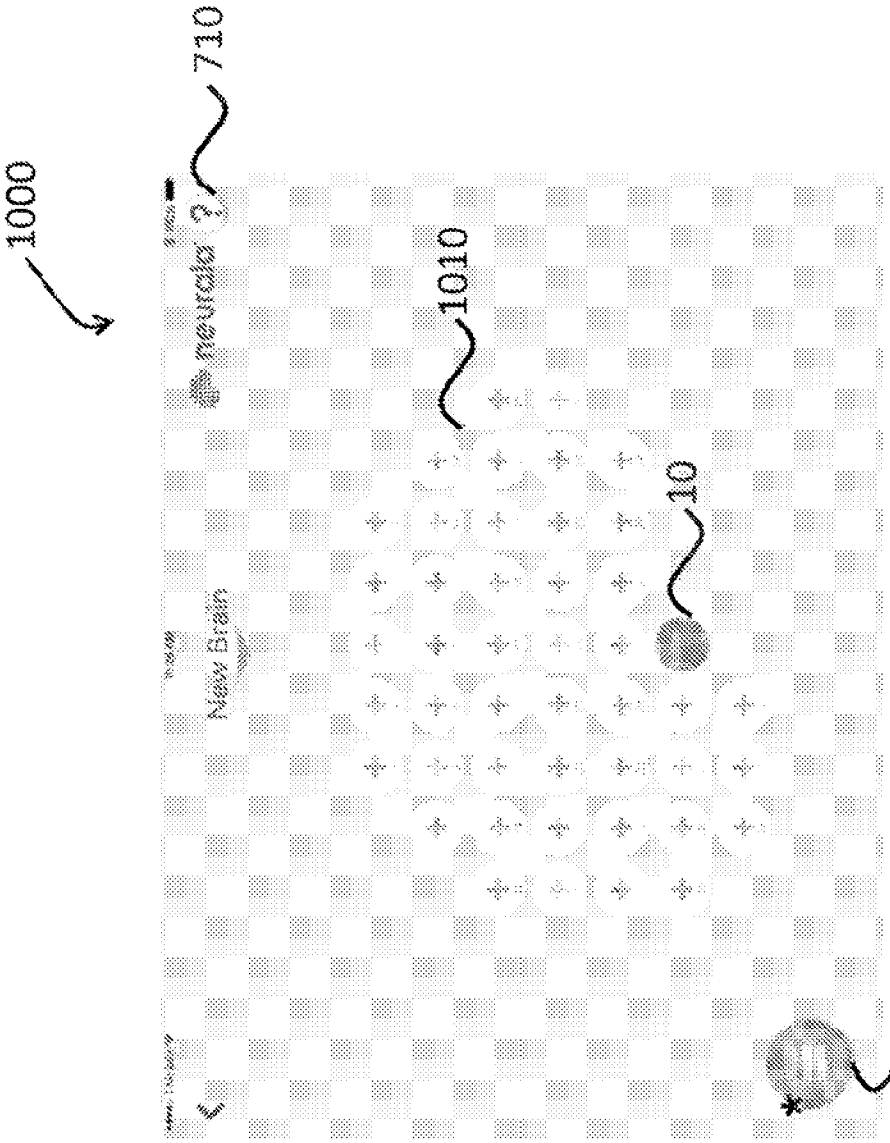


Fig. 10C

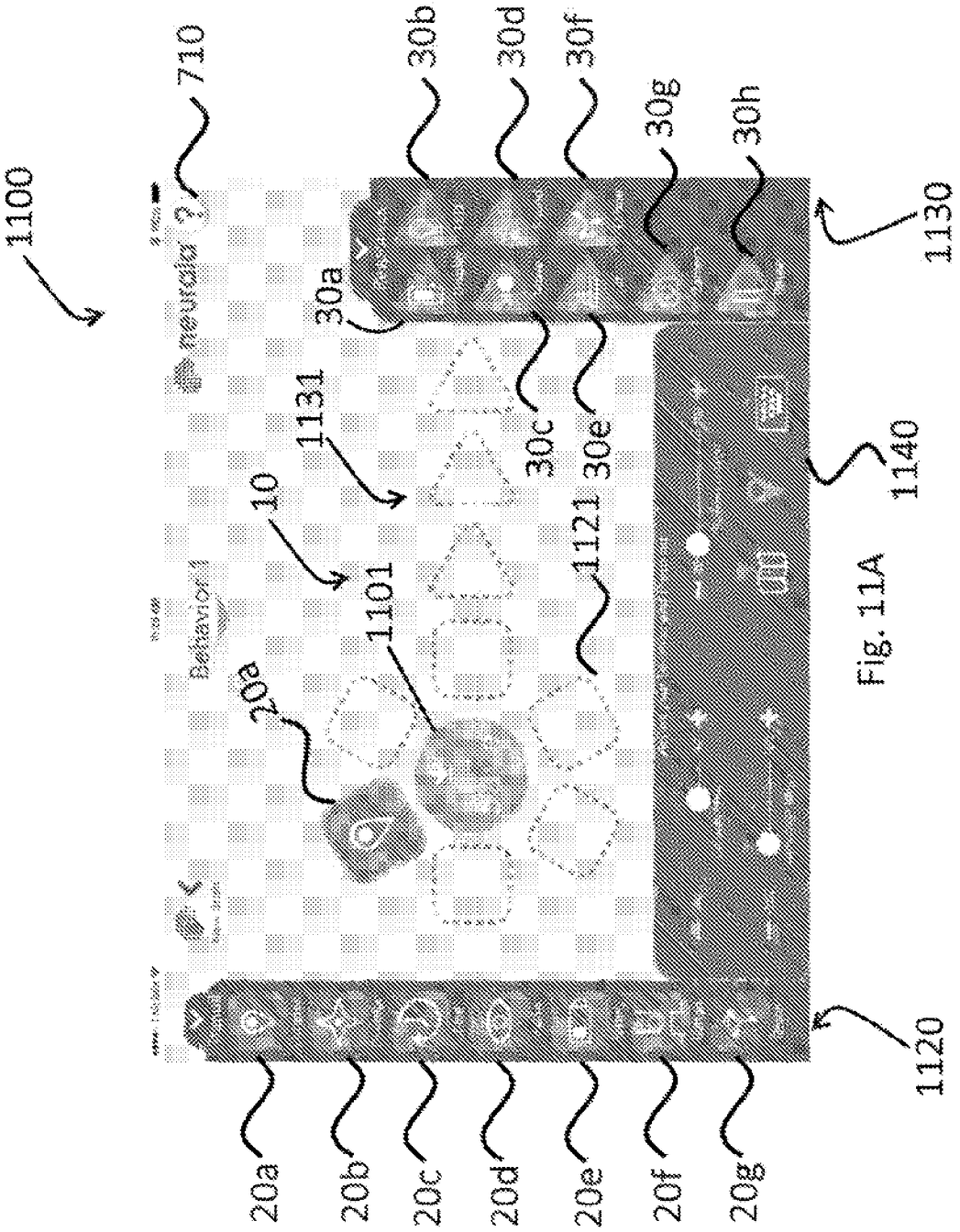

1100
710
30b
30d
30f
30g
30h
1130
30a
1131
10
30c
30e
1140
1121
1101
20a
1120
20a
20b
20c
20d
20e
20f
20g

Fig. 11A

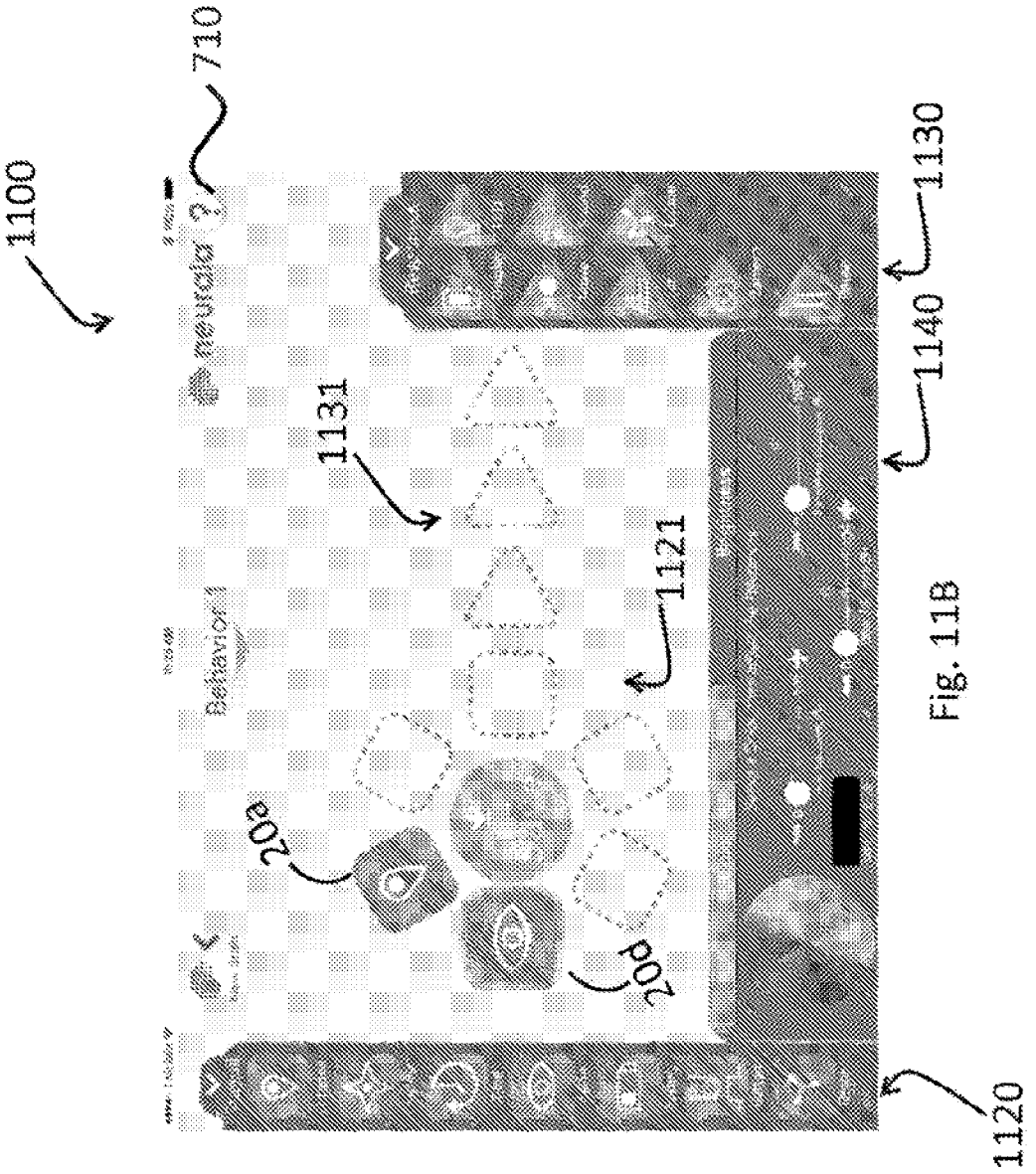


Fig. 11B

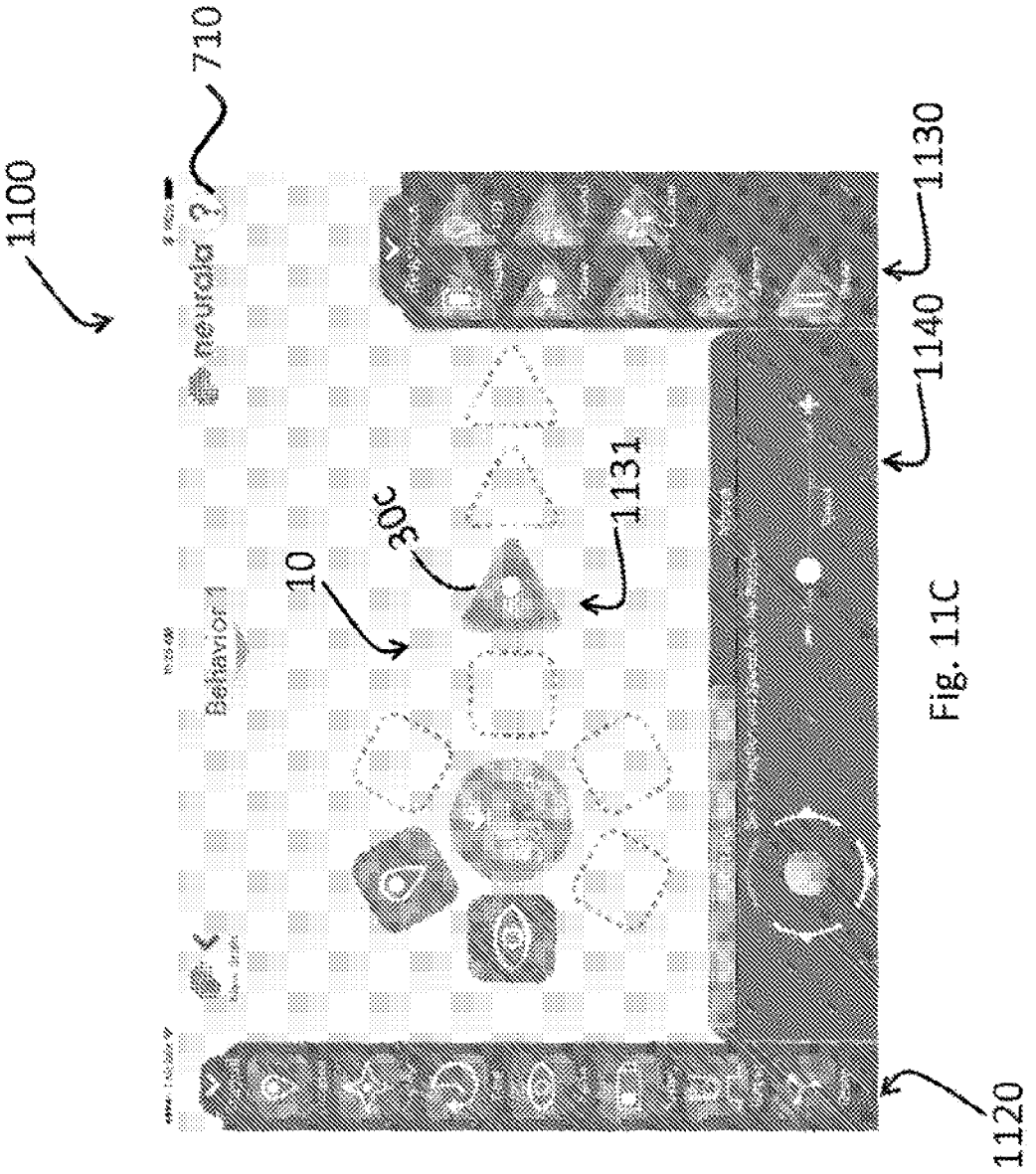


Fig. 11C

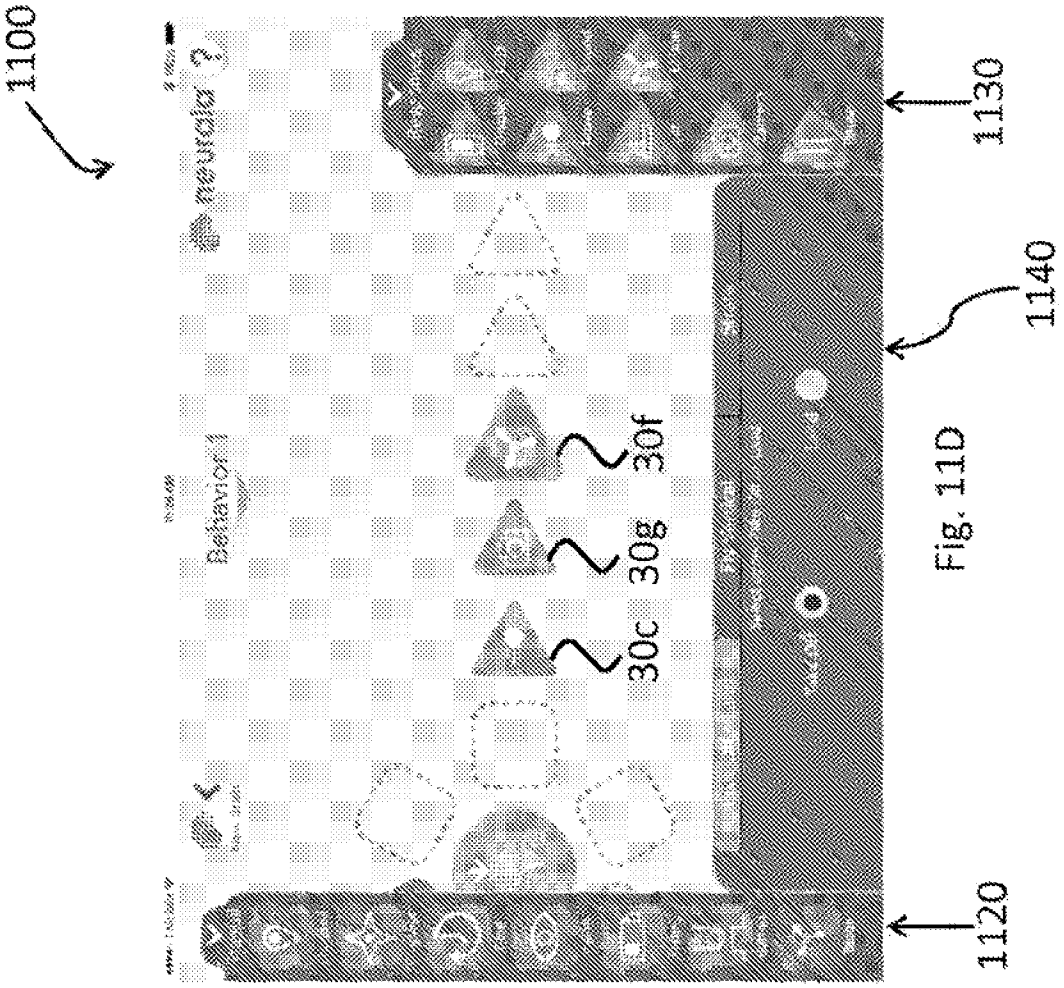


Fig. 11D

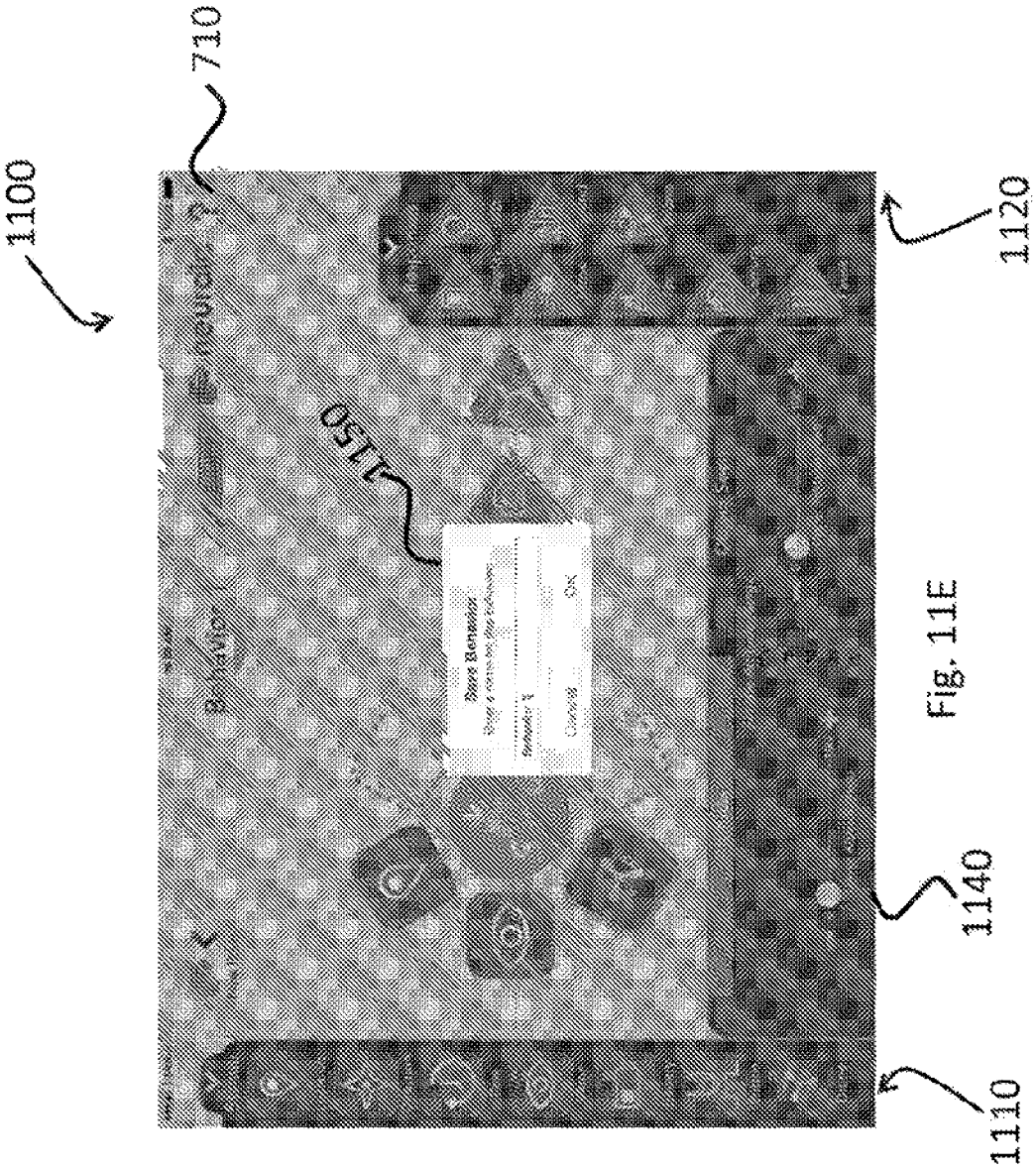


Fig. 11E

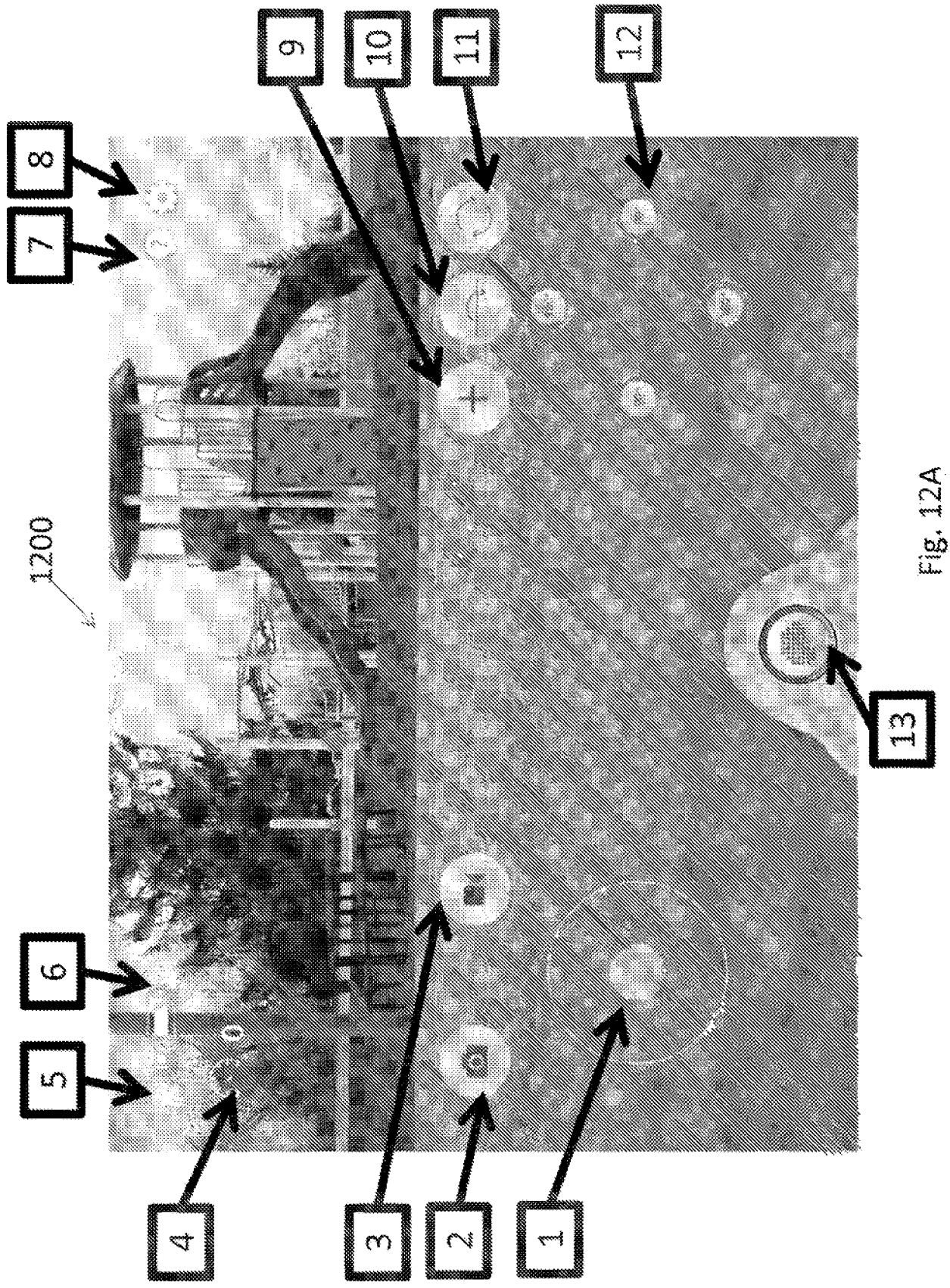


Fig. 12A

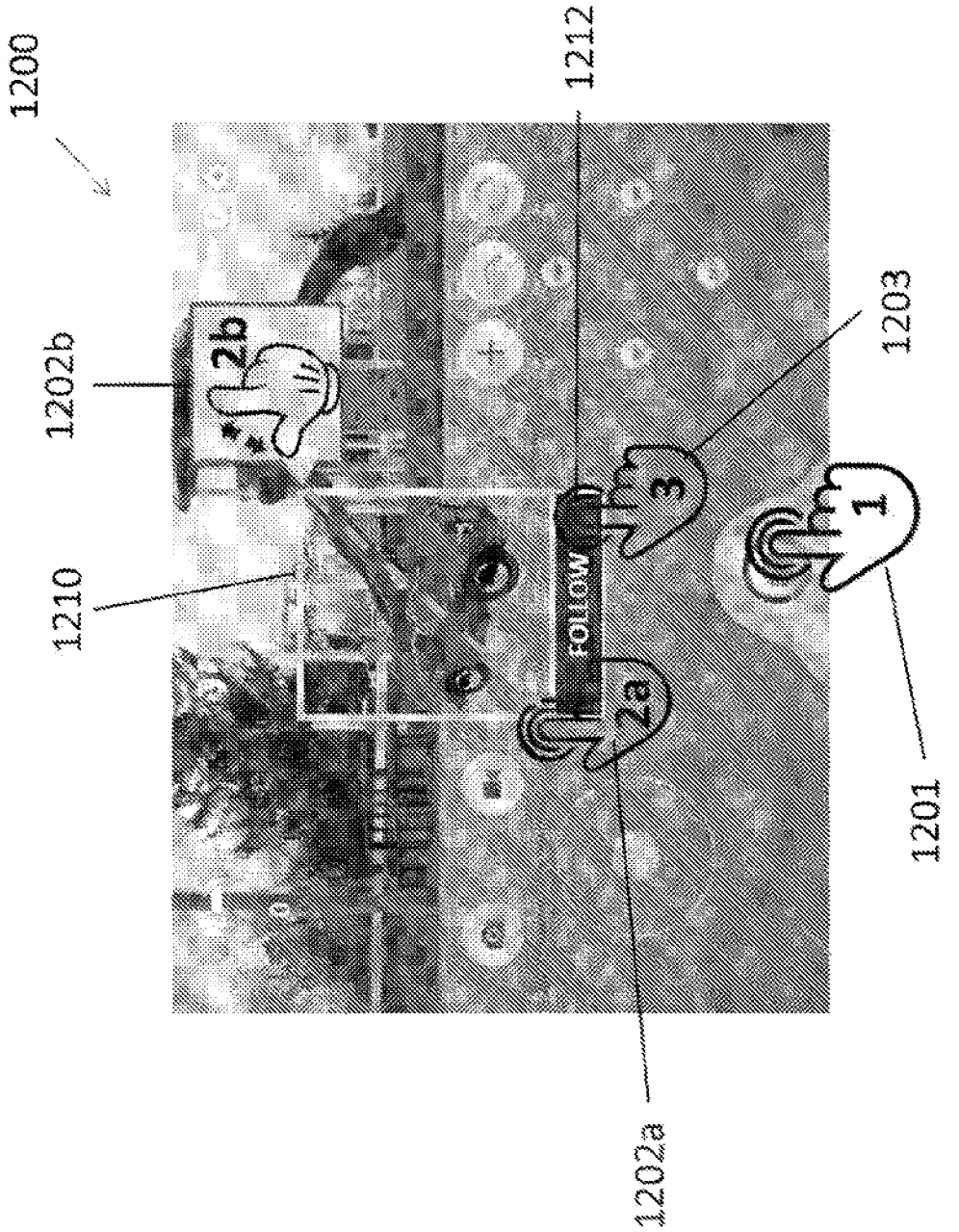


Fig. 12B

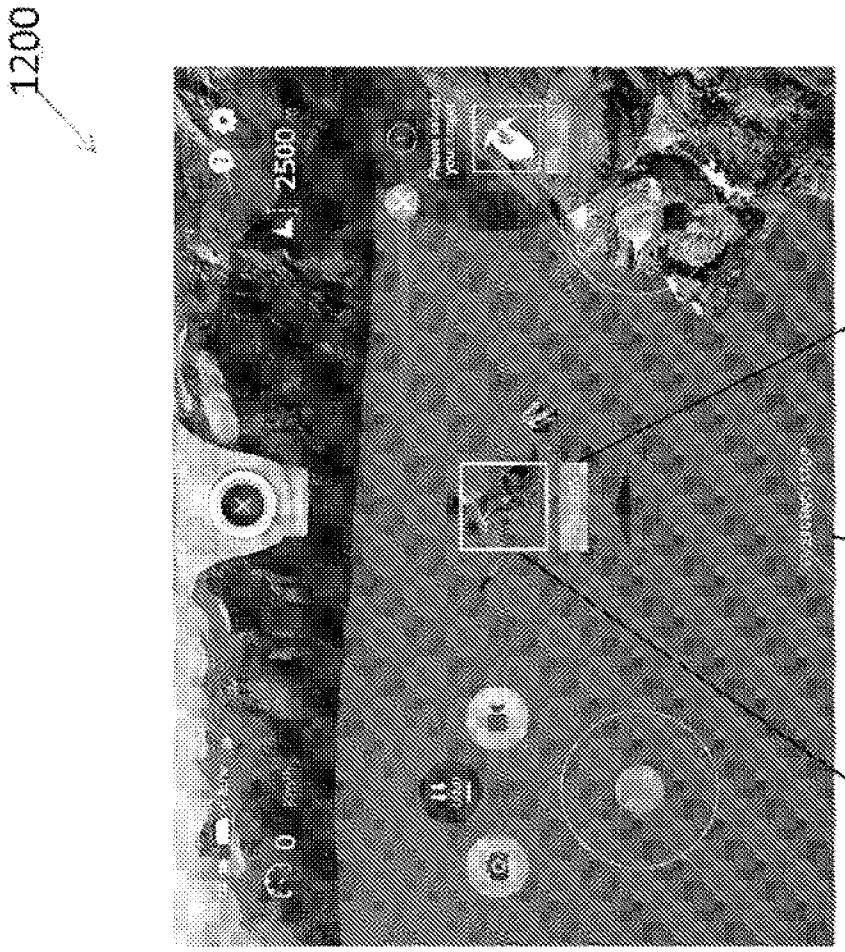


Fig. 12C

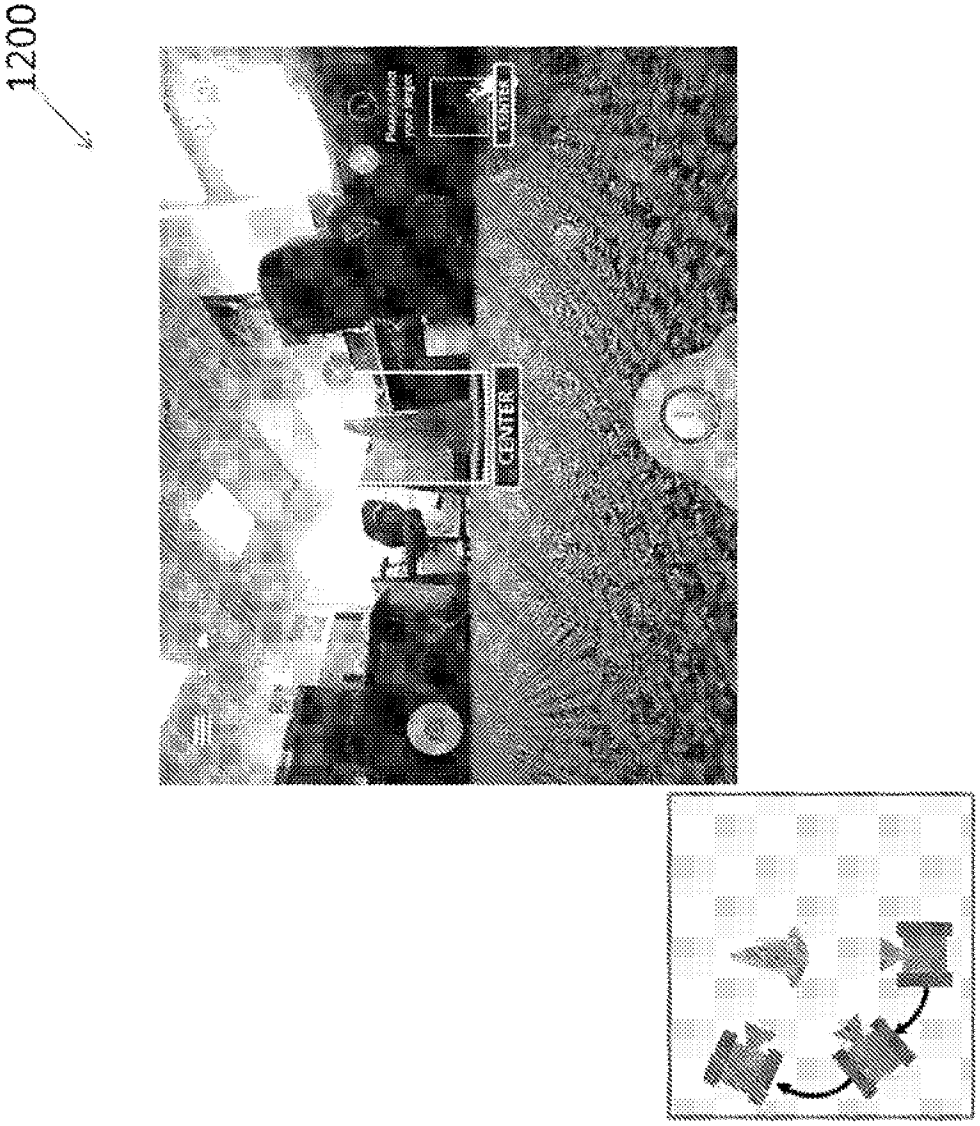
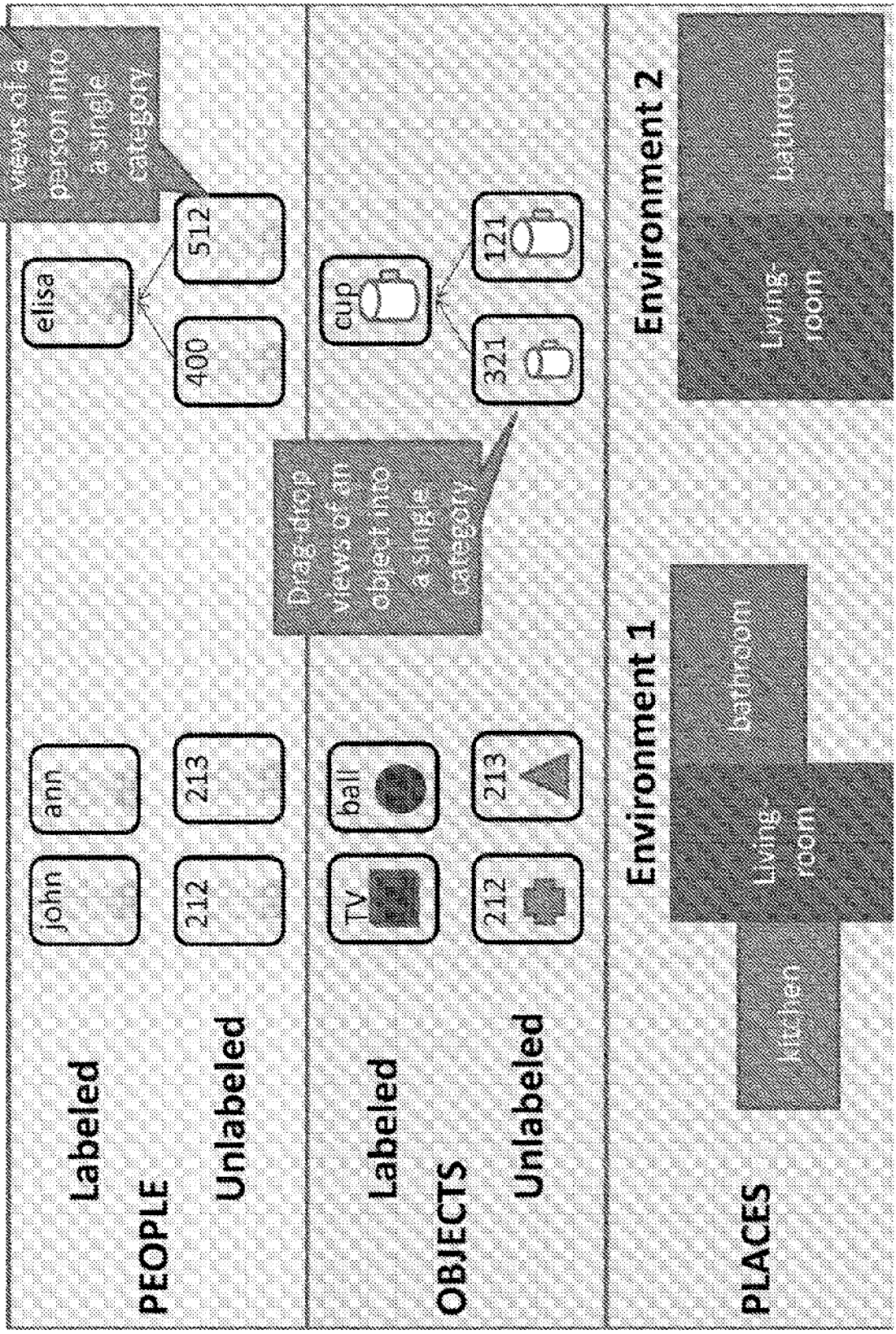


Fig. 12D

Fig. 13


Robot knowledge center
PEOPLE
Labeled
john
ann
Unlabeled
212
213
OBJECTS
Labeled
TV
ball
Unlabeled
212
213
PLACES
Environment 1
Living room
Environment 2
Living room
elisa
400
512
cup
321
121

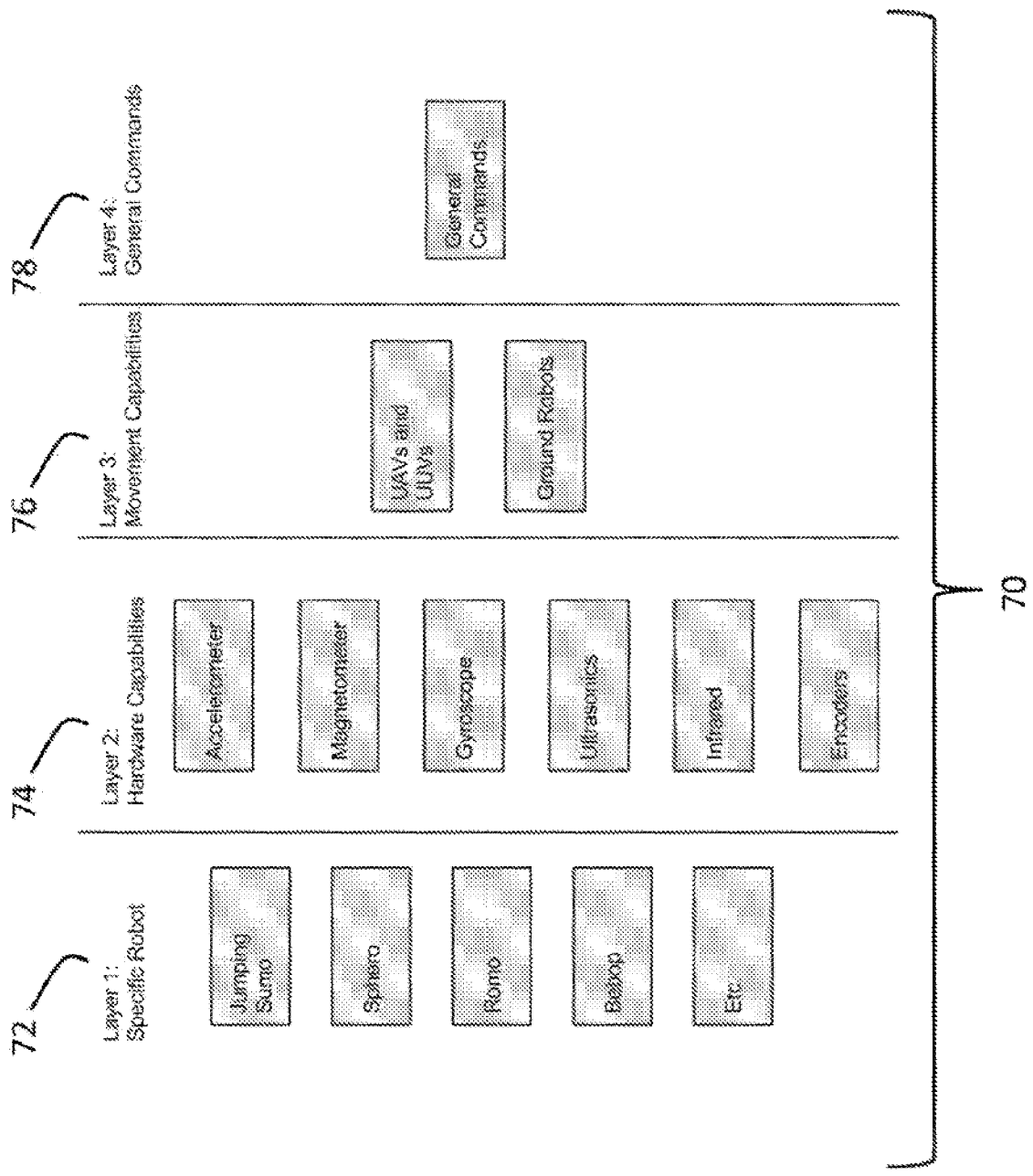
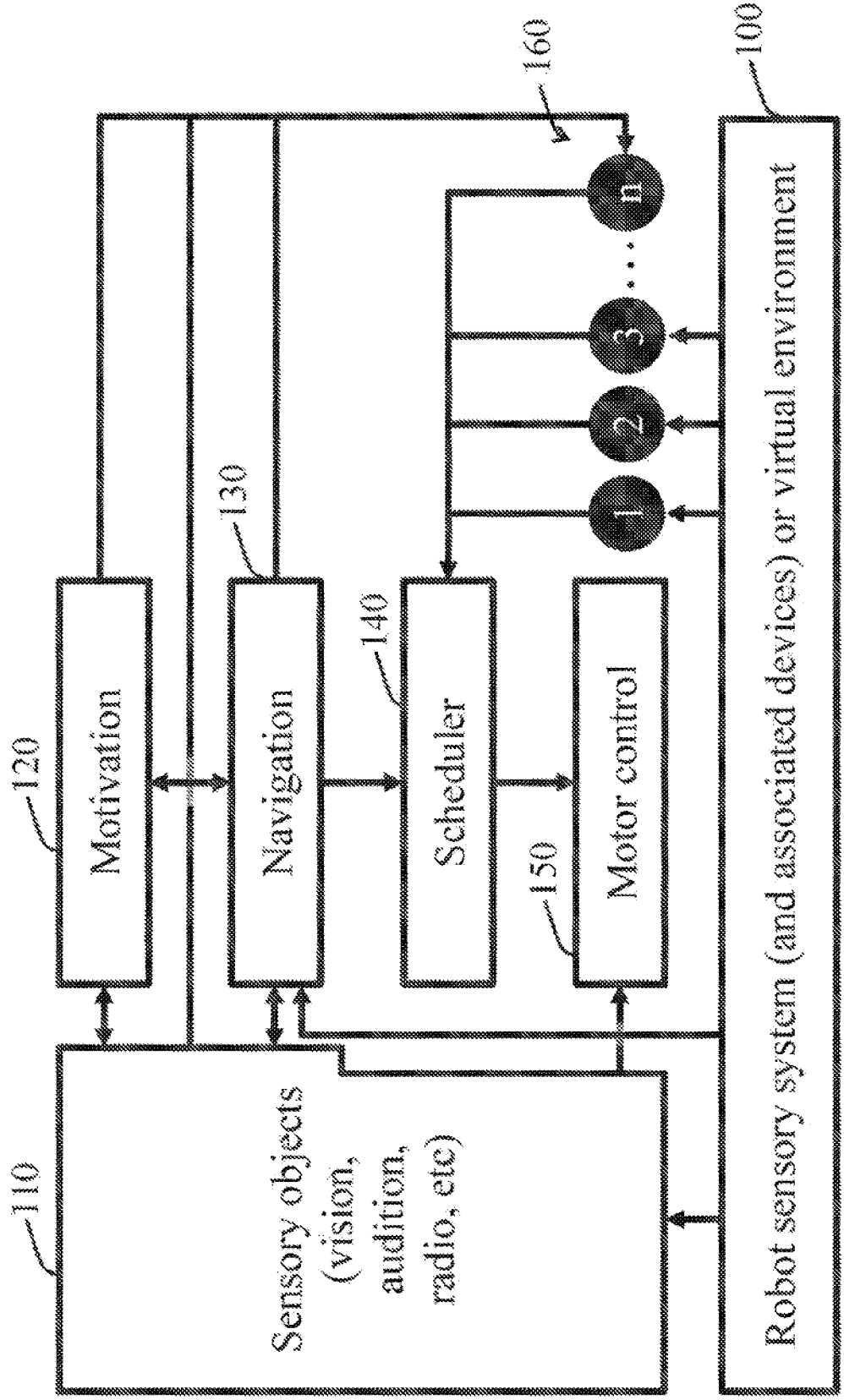

72
Layer 1:
Specific Robot
Jumping Sumo
Sphero
Rono
Bebop
Etc
74
Layer 2:
Hardware Capabilities
Accelerometer
Magnetometer
Gyroscope
Ultrasonics
Infrared
Encoders
76
Layer 3:
Movement Capabilities
UAVs and UUVs
Ground Robots
78
Layer 4:
General Commands
General Commands
70

Fig. 14

Fig. 15A


110
Sensory objects (vision, audition, radio, etc)
120
Motivation
130
Navigation
140
Scheduler
150
Motor control
160
1
2
3
n
100
Robot sensory system (and associated devices) or virtual environment

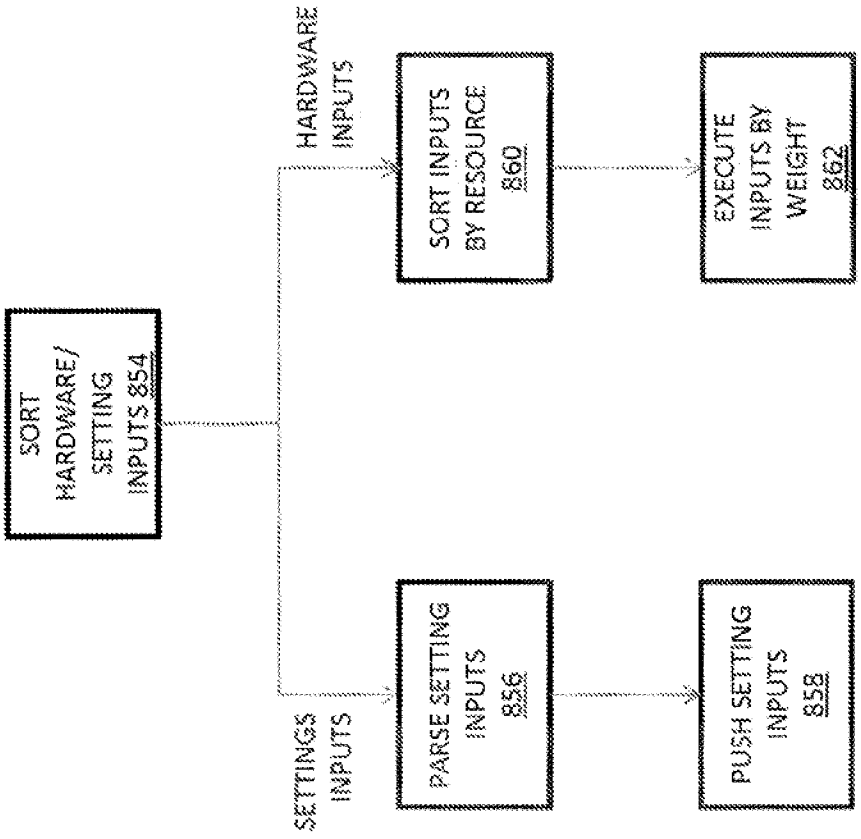

SORT HARDWARE/ SETTING INPUTS 854
HARDWARE INPUTS
SETTINGS INPUTS
SORT INPUTS BY RESOURCE 860
EXECUTE INPUTS BY WEIGHT 862
PARSE SETTING INPUTS 856
PUSH SETTING INPUTS 858

FIG. 15B

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD
First Named Inventor/Applicant Name:	Anatoli Gorchetchnikov
Filer:	Christopher M. Colice/Lindsey Lovato
Attorney Docket Number:	NRLA-001/03US 320730-2040

Filed as Small Entity

Filing Fees for Reissue (Utility)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
UTILITY REISSUE BASIC	2014	1	140	140
DESIGN AND UTILITY REISSUE BASIC	2114	1	300	300
DESIGN AND UTILITY REISSUE BASIC	2314	1	1080	1080
Pages:				
Claims:				
REISSUE CLAIMS IN EXCESS OF 20 FOR SMALL	2205	116	40	4640
REISSUE- INDEPENDENT CLAIMS	2204	13	210	2730
Miscellaneous-Filing:				

Adjustment date: 12/22/2017 SDIRETA1
11/13/2017 INTEFSW 00001030 501283 15808201
03 FC:2204 2730.00 CR

12/22/2017 SDIRETA1 00000007 501283 15808201
01 FC:2204 2520.00 DA

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/808,201	11/09/2017	Anatoli Gorchetchnikov	NRLA-001/03US 320730-2040

CONFIRMATION NO. 1077

58249
COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue, NW
Suite 700
Washington, DC 20004

FORMALITIES LETTER

OC000000096288042

Date Mailed: 12/26/2017

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

An application number and filing date have been accorded to this application. The application is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given TWO MONTHS from the date of this Notice within which to correct the informalities indicated below. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

The required item(s) identified below must be timely submitted to avoid abandonment:

- An abstract of the technical disclosure preferably not exceeding 150 words in length and commencing on a separate sheet in compliance with 37 CFR 1.72(b) is required. An abstract was not provided for this application.

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be mailed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web, including a copy of this Notice and selecting the document description "Applicant response to Pre-Exam Formalities Notice". https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html

For more information about EFS-Web please call the USPTO Electronic Business Center at 1-866-217-9197 or visit our website at http://www.uspto.gov/ebc.

If you are not using EFS-Web to submit your reply, you must include a copy of this notice.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/vostuart/

NVIDIA Ex1002 Page 468 of 1081

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 15/808,201

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	140		N/A	
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	300		N/A	
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	1080		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	136 minus 20 =	* 116	x 40 =	4640	OR		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	15 minus 3 =	* 12	x 210 =	2520			
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			0.00			
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				0.00			
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	8680		TOTAL	

APPLICATION AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
Application Size Fee (37 CFR 1.16(s))									
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE($)	ADDITIONAL FEE($)		RATE($)	ADDITIONAL FEE($)
Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
Application Size Fee (37 CFR 1.16(s))									
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
15/808,201	11/09/2017	2611	8680	NRLA-001/03US 320730-2040	136	15

CONFIRMATION NO. 1077

FILING RECEIPT

58249
COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue, NW
Suite 700
Washington, DC 20004

OC000000098288041

Date Mailed: 12/26/2017

Receipt is acknowledged of this reissue patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Anatoli Gorchetchnikov, Belmont, MA;
Heather Marie Ames, Milton, MA;
Massimiliano Versace, Milton, MA;
Fabrizio Santini, Jamaica Plain, MA;

Applicant(s)

Neurala, Inc., Boston, MA, Assignee (with 37 CFR 1.172 Interest);

Power of Attorney: The patent practitioners associated with Customer Number 58249

Domestic Priority data as claimed by applicant

This application is a REI of 14/147,015 01/03/2014 PAT 9189828
which is a CON of 11/860,254 09/24/2007 PAT 8648867
which claims benefit of 60/826,892 09/25/2006

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: No

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 12/25/2017

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 15/808,201**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No
**** SMALL ENTITY ****
Title

GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD

Preliminary Class

345

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

First Inventor:	Gorchetchnikov	Confirmation No.:	1077
Application No.:	15/808,201	Group Art Unit:	N/A
Filed:	November 9, 2017	Examiner:	Not Yet Assigned

For: GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD

MS Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO NOTICE TO FILE CORRECTED APPLICATION PAPERS

In response to the Notice to File Corrected Application Papers mailed December 26, 2017, enclosed is:

[X] Abstract

Issuance of a corrected filing receipt is respectfully requested.

and the payment of the following fee(s):

The Commissioner is hereby authorized to charge any appropriate fees under 37 C.F.R. §§1.16, 1.17, and 1.21 that may be required by this paper, and to credit any overpayment, to Deposit Account No. 50-1283.

Dated: January 3, 2018

Respectfully submitted,
COOLEY LLP

By: /Christopher Max Colice/
Christopher Max Colice
Reg. No. 65,634

COOLEY LLP
ATTN: IP Docketing Department
1299 Pennsylvania Avenue NW, Suite 700
Washington, DC 20004

Tel: (617) 937-2300
Fax: (202) 842-7899

159292254 v1

Electronic Acknowledgement Receipt

EFS ID:	31392051
Application Number:	15808201
International Application Number:	
Confirmation Number:	1077
Title of Invention:	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD
First Named Inventor/Applicant Name:	Anatoli Gorchetchnikov
Customer Number:	58249
Filer:	Christopher M. Colice/Lindsey Lovato
Filer Authorized By:	Christopher M. Colice
Attorney Docket Number:	NRLA-001/03US 320730-2040
Receipt Date:	03-JAN-2018
Filing Date:	09-NOV-2017
Time Stamp:	14:20:59
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Abstract	NRLA00103US-ABSTRACT.pdf	1116042 3e8a679e9272ae29698068055973e6b4cd801558	no	1

Warnings:

Information:					
2	Applicant Response to Pre-Exam Formalities Notice	NRLA00103US-ResponseNTFCAP.pdf	213200 ea5a2d5b325e2d322ebb3f7560045f6263b72dbd	no	1
Warnings:					
Information:					
			Total Files Size (in bytes):	1329242	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

US009189828B2

(12) **United States Patent**
Gorchetchnikov et al.

(10) **Patent No.: US 9,189,828 B2**
(45) **Date of Patent: *Nov. 17, 2015**

(54) **GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD**

(71) Applicant: **Neurala Inc.**, Boston, MA (US)

(72) Inventors: **Anatoli Gorchetchnikov**, Boston, MA (US); **Heather Marie Ames**, South Boston, MA (US); **Massimiliano Versace**, South Boston, MA (US); **Fabrizio Santini**, Jamaica Plain, MA (US)

(73) Assignee: **Neurala, Inc.**, Boston, MA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **14/147,015**

(22) Filed: **Jan. 3, 2014**

(65) **Prior Publication Data**

US 2014/0192073 A1 Jul. 10, 2014

Related U.S. Application Data

(63) Continuation of application No. 11/860,254, filed on Sep. 24, 2007, now Pat. No. 8,648,867.

(60) Provisional application No. 60/826,892, filed on Sep. 25, 2006.

(51) **Int. Cl.**

G06T 1/60	(2006.01)
G06F 9/50	(2006.01)
G06T 1/20	(2006.01)

(52) **U.S. Cl.**

CPC ***G06T 1/60*** (2013.01); ***G06F 9/5027*** (2013.01); ***G06T 1/20*** (2013.01); *G06F 2209/509* (2013.01)

(58) **Field of Classification Search**

CPC G06T 1/60

USPC 345/501, 503, 530, 531, 535; 348/739; 712/225

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,388,206	A	2/1995	Poulton et al.	
7,861,060	B1 *	12/2010	Nickolls et al.	712/22
8,648,867	B2 *	2/2014	Gorchetchnikov et al.	345/501
2005/0166042	A1	7/2005	Evans	
2007/0052713	A1	3/2007	Chung et al.	
2007/0279429	A1	12/2007	Ganzer	

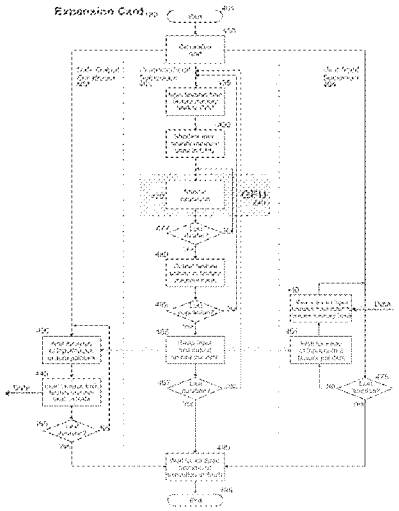
* cited by examiner

Primary Examiner — Maurice L. McDowell, Jr.
(74) *Attorney, Agent, or Firm* — Cooley LLP

(57) **ABSTRACT**

An accelerator system is implemented on an expansion card comprising a printed circuit board having (a) one or more graphics processing units (GPUs), (b) two or more associated memory banks (logically or physically partitioned), (c) a specialized controller, and (d) a local bus providing signal coupling compatible with the PCI industry standards. The controller handles most of the primitive operations to set up and control GPU computation. Thus, the computer's central processing unit (CPU) can be dedicated to other tasks. In this case a few controls (simulation start and stop signals from the CPU and the simulation completion signal back to CPU), GPU programs and input/output data are exchanged between CPU and the expansion card. Moreover, since on every time step of the simulation the results from the previous time step are used but not changed, the results are preferably transferred back to CPU in parallel with the computation.

20 Claims, 5 Drawing Sheets



PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 15/808,201

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A	150		N/A	
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	330		N/A	
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	1100		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	136 minus 20 =	* 116	x 50 =	5800	OR		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	15 minus 3 =	* 12	x 230 =	2760			
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			0.00			
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))				0.00			
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL	10140		TOTAL	

APPLICATION AS AMENDED - PART II

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
15/808,201	11/09/2017	2611	8680	NRLA-001/03US 320730-2040	136	15

CONFIRMATION NO. 1077

UPDATED FILING RECEIPT

58249
COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue, NW
Suite 700
Washington, DC 20004

OC000000097651881

Date Mailed: 02/27/2018

Receipt is acknowledged of this reissue patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Inventor(s)

Anatoli Gorchetchnikov, Belmont, MA;
Heather Marie Ames, Milton, MA;
Massimiliano Versace, Milton, MA;
Fabrizio Santini, Jamaica Plain, MA;

Applicant(s)

Neurala, Inc., Boston, MA, Assignee (with 37 CFR 1.172 Interest);

Power of Attorney: The patent practitioners associated with Customer Number 58249

Domestic Priority data as claimed by applicant

This application is a REI of 14/147,015 01/03/2014 PAT 9189828
which is a CON of 11/860,254 09/24/2007 PAT 8648867
which claims benefit of 60/826,892 09/25/2006

Foreign Applications for which priority is claimed (You may be eligible to benefit from the **Patent Prosecution Highway** program at the USPTO. Please see http://www.uspto.gov for more information.) - None.
Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: No

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

NVIDIA Ex1002 Page 478 of 1081

If Required, Foreign Filing License Granted: 12/25/2017

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 15/808,201**

Projected Publication Date: None, application is not eligible for pre-grant publication

Non-Publication Request: No

Early Publication Request: No
**** SMALL ENTITY ****
Title

GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD

Preliminary Class

345

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

PTO/SB/08a (07-09)
Approved for use through 07/31/2016. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT

(Use as many sheets as necessary)

Complete if Known	
Application Number	15/808,201
Filing Date	11-09-2017
First Named Inventor	Anatoli Gorchetchnikov
Art Unit	3992
Examiner Name	Not Yet Assigned
Attorney Docket Number	NRLA-001/03US 320730-2040

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code2 (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	1.	5,063,603	11-05-1991	Burt	
	2.	6,018,696	01-25-2000	Matsuoka et al.	
	3.	8,583,286	11-12-2013	Fleischer et al.	
	4.	2002/0064314	05-30-2002	Comaniciu et al.	
	5.	2002/0168100	11-14-2002	Woodall	
	6.	2003/0026588	02-06-2003	Elder et al.	
	7.	2009/0080695	03-26-2009	Yang et al.	
	8.	2009/0116688	05-07-2009	Monacos et al.	
	9.	2011/0279682	11-17-2011	Li et al.	
	10.	2013/0126703	05-23-2013	Caulfield	
	11.	2013/0131985	05-23-2013	Weiland et al.	
	12.	2015/0224648	08-13-2015	Lee et al.	

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	13.	Non-Final Office Action mailed 01/04/2018 from U.S. Application Serial No. 15/262,637, 23 pages (Atty. Docket No. NRLA-004/01US)	
	14.	WU, YAN & J. CAI, H. (2010). A Simulation Study of Deep Belief Network Combined with the Self-Organizing Mechanism of Adaptive Resonance Theory. 10.1109/CISE.2010.5677265, 4 pages.	

167642600 v1

Examiner Signature:		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

Electronic Acknowledgement Receipt

EFS ID:	31928206
Application Number:	15808201
International Application Number:	
Confirmation Number:	1077
Title of Invention:	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD
First Named Inventor/Applicant Name:	Anatoli Gorchetchnikov
Customer Number:	58249
Filer:	Nisha Somnath/Donna Doyle
Filer Authorized By:	Nisha Somnath
Attorney Docket Number:	NRLA-001/03US 320730-2040
Receipt Date:	01-MAR-2018
Filing Date:	09-NOV-2017
Time Stamp:	17:26:50
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	NRLA00103US-IDS.pdf	88913 b99e57f5e2904c312b7741fcd22103299a8b338e	no	2

Warnings:

Information:					
2	Information Disclosure Statement (IDS) Form (SB08)	NRLA00103US-SB08.pdf	184255 92348a18a5444612548 4f740091a9b730490d70d	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Non Patent Literature	WuASimulationStudy.pdf	3063358 bb49a12a2aaca67019cb80029fecad40e32160d9	no	4
Warnings:					
Information:					
		Total Files Size (in bytes):	3336526		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Attorney Docket No.: NRLA-001/03US 320730-2040 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	Anatoli Gorchetchnikov et al.	Confirmation No.:	1077
Application No.:	15/808,201	Group Art Unit:	3992
Filed:	November 9, 2017	Examiner:	Not Yet Assigned

FOR: **GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD**

VIA EFS or
Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. §1.97(b)

In accordance with the duty of disclosure set forth in 37 C.F.R. §1.56, Applicant(s) hereby submits the following information in conformance with 37 C.F.R. §§1.97 and 1.98.

☒ Pursuant to 37 C.F.R. §1.98, a copy of each non-US patent document cited in the attached Form PTO/SB/08 is enclosed.

☒ No copies of any U.S. Office Actions cited on the accompanying Form PTO/SB/08 are being provided because they are available on PAIR. Copies will be provided to the Examiner upon request.

☐ No copies of the publications listed on the attached Form PTO/SB/08 are being provided pursuant to 37 C.F.R. §1.98(d) because the publications were previously cited by or submitted to the Office in prior Application Serial No. _____ to which the above-identified application claims priority under 35 U.S.C. §120.

☒ No copies of any U.S. patents or U.S. patent application publications listed on the attached Form PTO/SB/08 are being provided pursuant to 37 C.F.R. §1.98.

☐ Publication(s) _____ listed on the attached Form PTO/SB/08 were cited in a foreign search or examination report corresponding to _____ application serial no. _____ and mailed on _____.

☐ Enclosed is a copy of a non-English publication(s) _____ Pursuant to §609 of the M.P.E.P., Applicant submits the attached foreign search or examination report, which cites such non-English language publication(s).

☐ Enclosed is a copy of a non-English publication(s) _____ English language publication _____ (copy enclosed) claims priority from this non-English publication.

☐ Enclosed is an explanation of non-English publication(s) _____ for which an English translation is not available.

☐ Enclosed is an English translation of non-English publication(s) _____ cited in the attached Form PTO/SB/08.

☐ Enclosed is a copy of pending patent Application Serial No. _____.

This Information Disclosure Statement is filed within any one of the following time periods:

☐ within three months from the filing date of this national application other than a CPA under 37 C.F.R. § 1.53(d);

☐ within three months from the date of entry of the national stage as set forth in 37 C.F.R. §1.491 in this international application;

☒ before the mailing date of a first office action on the merits; or

☐ before the mailing of a first office action after the filing of a request for continued examination under 37 C.F.R. § 1.114.

It is respectfully requested that the Examiner consider the above-noted information and return an initialed copy of the attached Form PTO/SB/08 to the undersigned.

Dated: 3/1/2018

Respectfully submitted,
COOLEY LLP

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW, Suite 700
Washington, DC 20004

Tel: (617) 937-2412
Fax: (202) 842-7899

By: /Nisha Somnath/
Nisha Somnath
Reg. No. L1194

167645567 v1

PTO/SB/08a (07-09)
Approved for use through 07/31/2016. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT LIST BY APPLICANT (Use as many sheets as necessary)	Complete if Known	
	Application Number	15/808,201
	Filing Date	11-09-2017
	First Named Inventor	Anatoli Gorchetchnikov
	Art Unit	3992
	Examiner Name	WOOD, WILLIAM H.
	Attorney Docket Number	NRLA-001/03US 320730-2040

NON PATENT LITERATURE DOCUMENTS

Examiner's Initials	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title ofthe item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	T[2]
	1.	AL-KAYSI, A. M. et al., A Multichannel Deep Belief Network for the Classification of EEG Data, from Ontology-based Information Extraction for Residential Land Use Suitability: A Case Study of the City of Regina, Canada, DOI 10.1007/978-3-319-26561-2_5, 8 pages (November 2015).	
	2.	APOLLONI, B. et al., Training a network of mobile neurons, Proceedings of International Joint Conference on Neural Networks, San Jose, CA, doi: 10.1109/IJCNN.2011.6033427, pp. 1683-1691 (July 31-August 5, 2011).	
	3.	BODDAPATI, V., Classifying Environmental Sounds with Image Networks, Thesis, Faculty of Computing Blekinge Institute of Technology, 37 pages (February 2017).	
	4.	KHALIGH-RAZAVI, S. -M. et al., Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation, PLoS Computational Biology, Vol. 10, Issue 11, 29 pages (November 2014).	
	5.	KIM, S., Novel approaches to clustering, biclustering and algorithms based on adaptive resonance theory and intelligent control, Doctoral Dissertations, Missouri University of Science and Technology, 125 pages (2016).	
	6.	Notice of Alllowance mailed May 22, 2018 from U.S. Application Serial No. 15/262,637, 6 pages (Atty. Docket No. NRLA-004/01US).	
	7.	REN, Y. et al., Ensemble Classification and Regression - Recent Developments, Applications and Future Directions, in IEEE Computational Intelligence Magazine, 10.1109/MCI.2015.2471235, 14 pages (2016).	
	8.	SUN, Z. et al., Recognition of SAR target based on multilayer auto-encoder and SNN, International Journal of Innovative Computing, Information and Control, Vol. 9, No. 11, pp. 4331-4341, November 2013.	

177153193 v1

Examiner Signature:		**Date Considered**	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] Applicant's unique citation designation number (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached. This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

Electronic Acknowledgement Receipt

EFS ID:	32712584
Application Number:	15808201
International Application Number:	
Confirmation Number:	1077
Title of Invention:	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD
First Named Inventor/Applicant Name:	Anatoli Gorchetchnikov
Customer Number:	58249
Filer:	Nisha Somnath/Donna Doyle
Filer Authorized By:	Nisha Somnath
Attorney Docket Number:	NRLA-001/03US 320730-2040
Receipt Date:	24-MAY-2018
Filing Date:	09-NOV-2017
Time Stamp:	20:54:57
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	320730-2040_Transmittal.pdf	92338 55a20fba694916bc8d43e310a2ffbe35354872a3	no	3

Warnings:

Information:					
2	Information Disclosure Statement (IDS) Form (SB08)	320730-2040_SB08.pdf	182961 c018bcbc3ac8eac99df0458457be4e9b697cc22a	no	1
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Non Patent Literature	Al-kaysi.pdf	6252969 18402380c08f7674fa0487dd8cc0182fc2b14f56	no	8
Warnings:					
Information:					
4	Non Patent Literature	Apolloni.pdf	6643199 b5e3d778d4a398bc92790475f150824f9a7283b0	no	10
Warnings:					
Information:					
5	Non Patent Literature	Boddapati.pdf	17682823 93b85d09ddbbe53163db27b51cd1000e4a0e2edc	no	37
Warnings:					
Information:					
6	Non Patent Literature	Kim_Part1.pdf	20716151 0d598283d9e82d7adab07eae9f5a071d1ed13c6c	no	32
Warnings:					
Information:					
7	Non Patent Literature	Kim_Part2.pdf	23091329 4f66ac1fd1df79828c3180025b975e64c288af9f	no	32
Warnings:					
Information:					
8	Non Patent Literature	Kim_Part3.pdf	23633172 9ce6046c3d306084d5c7177c635c63b2eed3131e	no	32
Warnings:					
Information:					

9	Non Patent Literature	Kim_Part4.pdf	21322257 ce74363b011c33bee42c7886b1129387675f1441	no	29
Warnings:					
Information:					
10	Non Patent Literature	Ren.pdf	16881485 be17eff83726c153614cd83a80aff8f903c5e4b2	no	14
Warnings:					
Information:					
11	Non Patent Literature	Sun.pdf	9259609 bceae7e826cb2c20ada3128ca10ca9a9d99621ba	no	11
Warnings:					
Information:					
12	Non Patent Literature	Khaligh-Razavi_reduced.pdf	8497649 244dd66f357e0a01acf60c5b4ce27d06e060bf0e	no	29
Warnings:					
Information:					
		Total Files Size (in bytes):	154255942		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	Anatoli Gorchetchnikov et al.	Confirmation No.:	1077
Application No.:	15/808,201	Group Art Unit:	3992
Filed:	November 9, 2017	Examiner:	WOOD, WILLIAM H.

FOR: **GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD**

VIA EFS or
Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. §1.97(b)

In accordance with the duty of disclosure set forth in 37 C.F.R. §1.56, Applicant(s) hereby submits the following information in conformance with 37 C.F.R. §§1.97 and 1.98.

☒ Pursuant to 37 C.F.R. §1.98, a copy of each non-US patent document cited in the attached Form PTO/SB/08 is enclosed.

☒ No copies of any U.S. Office Actions cited on the accompanying Form PTO/SB/08 are being provided because they are available on PAIR. Copies will be provided to the Examiner upon request.

☐ No copies of the publications listed on the attached Form PTO/SB/08 are being provided pursuant to 37 C.F.R. §1.98(d) because the publications were previously cited by or submitted to the Office in prior Application Serial No. _____ to which the above-identified application claims priority under 35 U.S.C. §120.

☐ No copies of any U.S. patents or U.S. patent application publications listed on the attached Form PTO/SB/08 are being provided pursuant to 37 C.F.R. §1.98.

☐ Publication(s) _____ listed on the attached Form PTO/SB/08 were cited in a foreign search or examination report corresponding to _____ application serial no. _____ and mailed on _____.

☐ Enclosed is a copy of a non-English publication(s) _____ Pursuant to §609 of the M.P.E.P., Applicant submits the attached foreign search or examination report, which cites such non-English language publication(s).

☐ Enclosed is a copy of a non-English publication(s) _____ English language publication _____ (copy enclosed) claims priority from this non-English publication.

☐ Enclosed is an explanation of non-English publication(s) _____ for which an English translation is not available.

☐ Enclosed is an English translation of non-English publication(s) _____ cited in the attached Form PTO/SB/08.

☐ Enclosed is a copy of pending patent Application Serial No. _____.

This Information Disclosure Statement is filed within any one of the following time periods:

☐ within three months from the filing date of this national application other than a CPA under 37 C.F.R. § 1.53(d);

☐ within three months from the date of entry of the national stage as set forth in 37 C.F.R. §1.491 in this international application;

☒ before the mailing date of a first office action on the merits; or

☐ before the mailing of a first office action after the filing of a request for continued examination under 37 C.F.R. § 1.114.

In accordance with 37 C.F.R. § 1.97(g), this Information Disclosure Statement shall not be construed as to mean that a search has been made.

In accordance with 37 C.F.R. § 1.97(h), the filing of this Information Disclosure Statement shall not be construed to be an admission that the information cited in the statement is, or is considered to be material to patentability as defined by 37 C.F.R § 1.56(b).

Remarks

It is respectfully requested that the Examiner consider the above-noted information and return an initialed copy of the attached Form used in lieu of PTO/SB/08 to the undersigned.

The Director is hereby authorized to charge any deficiency or credit any overpayment in the fees filed, asserted to be filed or which should have been filed herewith to our Deposit Account No. 50-1283 which the undersigned is authorized to draw.

The Examiner is invited to contact the undersigned by telephone if it is felt that a telephone interview would advance the prosecution of the present application.

Dated: 5/23/2018

Respectfully submitted,
COOLEY LLP

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW, Suite 700
Washington, DC 20004

Tel: (617) 937-2412
Fax: (202) 842-7899

By: /Nisha Somnath/
Nisha Somnath
Reg. No. L1194

177153502 v1

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 01/31/2018. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date
15/808,201	11/09/2017

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 149890

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

☐ Firm or Individual Name	
Address	
City	State / Zip
Country	
Telephone	Email

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

Neurala, Inc.

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	Date (Optional)
Name	Heather Anne Vierra
Title	COO

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☐ Total of forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Electronic Acknowledgement Receipt

EFS ID:	33352056
Application Number:	15808201
International Application Number:	
Confirmation Number:	1077
Title of Invention:	GRAPHIC PROCESSOR BASED ACCELERATOR SYSTEM AND METHOD
First Named Inventor/Applicant Name:	Anatoli Gorchetchnikov
Customer Number:	58249
Filer:	Christopher M. Colice/Lindsey Lovato
Filer Authorized By:	Christopher M. Colice
Attorney Docket Number:	NRLA-001/03US 320730-2040
Receipt Date:	02-AUG-2018
Filing Date:	09-NOV-2017
Time Stamp:	09:48:37
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	NRLA001US03-POA2.pdf	1685094 5e5e10d140441b2136494f16226b657af2882188	no	1

Warnings:

Information:	
Total Files Size (in bytes):	1685094

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/808,201	11/09/2017	Anatoli Gorchetchnikov	NRLA-001/03US 320730-2040

CONFIRMATION NO. 1077

POWER OF ATTORNEY NOTICE

58249
COOLEY LLP
ATTN: IP Docketing Department
1299 Pennsylvania Avenue, NW
Suite 700
Washington, DC 20004

OC000000101501039

Date Mailed: 08/07/2018

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 08/02/2018.

- The Power of Attorney to you in this application has been revoked by the applicant. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/zabraha/

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
15/808,201	11/09/2017	Anatoli Gorchetchnikov	NRLA-001/03US 320730-2040

CONFIRMATION NO. 1077

POA ACCEPTANCE LETTER

149890
SMITH BALUCH LLP
376 BOYLSTON ST
STE 401
BOSTON, MA 02116

OC000000101501067

Date Mailed: 08/07/2018

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 08/02/2018.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/zabraha/

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/808,201	11/09/2017	Anatoli Gorchetchnikov	NRLA-001US03	1077

149890 7590 11/20/2018

SMITH BALUCH LLP
376 BOYLSTON ST
STE 401
BOSTON, MA 02116

EXAMINER	
WOOD, WILLIAM H	

ART UNIT	PAPER NUMBER
3992	

NOTIFICATION DATE	DELIVERY MODE
11/20/2018	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docketing@smithbaluch.com
smith@smithbaluch.com

Office Action Summary	Application No. 15/808,201	Applicant(s) Gorchetchnikov et al.	
	Examiner WILLIAM H WOOD	Art Unit 3992	AIA Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☑ Responsive to communication(s) filed on 11/09/2017.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL.** 2b) ☑ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☑ Claim(s) 1-136 is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6) ☐ Claim(s) _____ is/are allowed.

7) ☑ Claim(s) 1-136 is/are rejected.

8) ☐ Claim(s) _____ is/are objected to.

9) ☐ Claim(s) _____ are subject to restriction and/or election requirement

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to **PPHfeedback@uspto.gov.**

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☑ The drawing(s) filed on 11/09/2017 is/are: a)☑ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☑ Notice of References Cited (PTO-892)

2) ☑ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date 12/08/2017; 03/01/2018; 05/24/2018.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date _____.

4) ☐ Other: _____.

The present application is being examined under the pre-AIA first to invent provisions.

DETAILED ACTION

For reissue applications filed before September 16, 2012, all references to 35 U.S.C. 251 and 37 CFR 1.172, 1.175, and 3.73 are to the law and rules in effect on September 15, 2012. Where specifically designated, these are "pre-AIA" provisions.

For reissue applications filed on or after September 16, 2012, all references to 35 U.S.C. 251 and 37 CFR 1.172, 1.175, and 3.73 are to the current provisions.

Claims 1-136 are currently pending in the application. Claims 1-20 are original claims to patent US 9,189,828 B2 and claims 21-136 are newly added claims.

Reissue

Applicant is reminded of the continuing obligation under 37 CFR 1.178(b), to timely apprise the Office of any prior or concurrent proceeding in which Patent No. 9,189,828 is or was involved. These proceedings would include interferences, reissues, reexaminations, and litigation.

Applicant is further reminded of the continuing obligation under 37 CFR 1.56, to timely apprise the Office of any information which is material to patentability of the claims under consideration in this reissue application.

These obligations rest with each individual associated with the filing and prosecution of this application for reissue. See also MPEP §§ 1404, 1442.01 and 1442.04.

Applicant is notified that any subsequent amendment to the specification and/or claims must comply with 37 CFR 1.173(b).

Information Disclosure Statement

The information disclosure statements (IDS) submitted on 12/08/2017, 03/01/2018, and 05/24/2018 have been considered in accordance with MPEP 609, including timing and content requirements (see also 37 C.F.R. 1.97 and 1.98). Documents not meeting a particular criteria are lined through and not considered.

Objections

The reissue oath/declaration filed with this application is defective and objected to because it fails: 1) to identify at least one error under 35 USC 251 (37 CFR 1.175(a)). The declaration fails to indicate the "antecedent basis errors" of claim 1 or how claim 21 is broadened.

The amended specification filed with this application is defective and objected to (see 37 CFR 1.173(b)) because of the following: "The matter to be omitted by reissue must be enclosed in brackets", not by strikethrough.

Claim Rejections - 35 USC § 112

The following is a quotation of 35 U.S.C. 112(b):
(b) CONCLUSION.—The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention.

The following is a quotation of 35 U.S.C. 112 (pre-AIA), second paragraph:
The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1-20, 90-100, 115, and 117-136 are rejected under 35 U.S.C. 112(b) or 35 U.S.C. 112 (pre-AIA), second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the inventor or a joint inventor, or for pre-AIA the applicant regards as the invention.

Claim 1 recites the limitation "by the at least one graphic processing unit" in the last element of the claim. There is insufficient antecedent basis for this limitation in the claim, as it should read "graphics".

Claim 11 recites the limitation "the central processor unit". There is insufficient antecedent basis for this limitation in the claim, as it should read "processing".

Claims 12-14 and 16-19 recite the limitations "the input data" and "the output data". There is insufficient antecedent basis for these limitations in the claims.

Claims 90 and 94 recite the limitation "the output data". There is insufficient antecedent basis for this limitation in the claims.

Claims 95 and 98-100 recite the limitations "the input data" and "the output data". There is insufficient antecedent basis for these limitations in the claims.

Claim 115 recites the limitation "the accelerator controller". There is insufficient antecedent basis for this limitation in the claim.

Claim 117 recites the limitation "the at least one graphic processing unit" in the last element of the claim. There is insufficient antecedent basis for this limitation in the claim.

Claim 127 recites the limitation "the central processor unit". There is insufficient antecedent basis for this limitation in the claim, as it should read "processing".

Claims 128-130 and 132-135 recite the limitations "the input data" and "the output data". There is insufficient antecedent basis for these limitations in the claims.

Claim Rejections – 35 USC § 251

Claims 1-136 are rejected under 35 U.S.C. 251 as the error statement of the oath/declaration is insufficient and must address: 1) "Any error in the claims must be identified by reference to the specific

claim(s) and the specific claim language wherein lies the error"; and 2) "the reissue oath or declaration must also identify a claim that the application seeks to broaden" (see MPEP 1414 II.).

Claims 21-116 are rejected under 35 U.S.C. 251 as being an improper recapture of broadened claimed subject matter surrendered in the application for the patent upon which the present reissue is based. See *Greenliant Systems, Inc. et al v. Xicor LLC*, 692 F.3d 1261, 103 USPQ2d 1951 (Fed. Cir. 2012); *In re Shahram Mostafazadeh and Joseph O. Smith*, 643 F.3d 1353, 98 USPQ2d 1639 (Fed. Cir. 2011); *North American Container, Inc. v. Plastipak Packaging, Inc.*, 415 F.3d 1335, 75 USPQ2d 1545 (Fed. Cir. 2005); *Pannu v. Storz Instruments Inc.*, 258 F.3d 1366, 59 USPQ2d 1597 (Fed. Cir. 2001); *Hester Industries, Inc. v. Stein, Inc.*, 142 F.3d 1472, 46 USPQ2d 1641 (Fed. Cir. 1998); *In re Clement*, 131 F.3d 1464, 45 USPQ2d 1161 (Fed. Cir. 1997); *Ball Corp. v. United States*, 729 F.2d 1429, 1436, 221 USPQ 289, 295 (Fed. Cir. 1984). A broadening aspect is present in the reissue which was not present in the application for patent. The record of the application for the patent shows that the broadening aspect (in the reissue) relates to claimed subject matter that applicant previously surrendered during the prosecution of the application. Accordingly, the narrow scope of the claims in the patent was not an error within the meaning of 35 U.S.C. 251, and the broader scope of claim subject matter surrendered in the application for the patent cannot be recaptured by the filing of the present reissue application.

It is noted that the following is the three step test for determining recapture in reissue applications (see: MPEP 1412.02(II)):

"(1) first, we determine whether, and in what respect, the reissue claims are broader in scope than the original patent claims;

(2) next, we determine whether the broader aspects of the reissue claims relate to subject matter surrendered in the original prosecution; and

(3) finally, we determine whether the reissue claims were materially narrowed in other respects, so that the claims may not have been enlarged, and hence avoid the recapture rule."

(Step 1: MPEP 1412.02(II.)(A)) In the instant reissue application, of patent US 9,189,828 B2, by way of the preliminary amendment:

a. Patent Owner seeks to broaden newly added independent claim 21 in this reissue at least by deleting/omitting the limitations: a central processing unit; main memory; and an accelerator, "operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory"; "intermediate computations in the sequence of computations yielding intermediate results"; and to transfer at least a portion of the output data from the accelerator memory "to the main memory" (see issued claims 1).

b. Patent Owner seeks to broaden newly added independent claim 42 in this reissue at least by deleting/omitting the limitations: "on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory"; (A) performing, by the GPU, "the sequence of computations on a first portion of the input data so as to generate a first portion of the output data"; "intermediate computations in the sequence of computations yielding intermediate results"; "(B) in parallel with performing the sequence of computations by the GPU in (A)"; transferring a second portion of the input data "from the main memory to the accelerator via the bus"; and "(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus" (see issued claim 12).

c. Patent Owner seeks to broaden newly added independent claim 56 in this reissue at least by deleting/omitting the limitations: a central processing unit; main memory; and an accelerator, "operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory"; "intermediate computations in the sequence of computations yielding intermediate results"; accelerator memory, operably coupled to the at least one graphic processing unit, "to store the results of the plurality of sequential computations"; and "a controller, operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory and to transfer at least a portion of the output data from the accelerator memory to the main memory during performance of the sequence of computations by the at least one graphic processing unit" (see issued claims 1).

d. Patent Owner seeks to broaden newly added independent claim 62 in this reissue at least by deleting/omitting the limitations: "on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory"; "intermediate computations in the sequence of computations yielding intermediate results"; "(B) in parallel with performing the sequence of computations by the GPU in (A)"; transferring a second portion of the input data "from the main memory to the accelerator via the bus"; and "(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus" (see issued claim 12).

e. Patent Owner seeks to broaden newly added independent claim 68 in this reissue at least by deleting/omitting the limitations: "on a computer system comprising a … a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the

CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory"; "(A) performing, by the GPU, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data"; "intermediate computations in the sequence of computations yielding intermediate results"; "(B) in parallel with performing the sequence of computations by the GPU in (A) transferring a second portion of the input data from the main memory to the accelerator via the bus"; and "(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus" (see issued claim 12).

f. Patent Owner seeks to broaden newly added independent claim 79 in this reissue at least by deleting/omitting the limitations: "an accelerator, operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory"; "at least one graphics processing unit to perform a sequence of computations on the at least a portion of the input data so as to generate output data, intermediate computations in the sequence of computations yielding intermediate results"; "accelerator memory, operably coupled to the at least one graphic processing unit, to store the results of the plurality of sequential computations"; and "a controller, operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory and to transfer at least a portion of the output data from the accelerator memory to the main memory during performance of the sequence of computations by the at least one graphic processing unit" (see issued claims 1).

g. Patent Owner seeks to broaden newly added independent claim 90 in this reissue at least by deleting/omitting the limitations: "on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator

operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory"; and "via the bus" (see issued claim 12).

h. Patent Owner seeks to broaden newly added independent claim 95 in this reissue at least by deleting/omitting the limitations: a central processing unit; main memory; and "an accelerator, operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory"; "accelerator memory, operably coupled to the at least one graphic processing unit, to store the results of the plurality of sequential computations"; and "a controller, operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory and to transfer at least a portion of the output data from the accelerator memory to the main memory during performance of the sequence of computations by the at least one graphic processing unit" (see issued claims 1).

i. Patent Owner seeks to broaden newly added independent claim 101 in this reissue at least by deleting/omitting the limitations: "on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory"; "(A) performing, by the GPU, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data, intermediate computations in the sequence of computations yielding intermediate results"; "(B) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the input data from the main memory to the accelerator via the bus"; and "(C) in parallel with performing the sequence of computations by the GPU in

(A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus" (see issued claim 12).

j. Patent Owner seeks to broaden newly added independent claim 108 in this reissue at least by deleting/omitting the limitations: "on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory"; "(A) performing, by the GPU, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data, intermediate computations in the sequence of computations yielding intermediate results"; "(B) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the input data from the main memory to the accelerator via the bus"; and "(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus" (see issued claim 12).

k. Patent Owner seeks to broaden newly added independent claim 112 in this reissue at least by deleting/omitting the limitations: "an accelerator, operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory"; "intermediate computations in the sequence of computations yielding intermediate results"; accelerator memory, operably coupled to the at least one graphic processing unit, "to store the results of the plurality of sequential computations"; and "a controller, operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory and to transfer at least a portion of the output data from the accelerator memory to the main memory

during performance of the sequence of computations by the at least one graphic processing unit" (see issued claims 1).

l. Patent Owner seeks to broaden newly added independent claim 117 in this reissue at least by deleting/omitting the limitation: "graphics" processing unit (see issued claims 1).

m. Patent Owner seeks to broaden newly added independent claim 128 in this reissue at least by deleting/omitting the limitations: "graphics" processing unit (see issued claims 12).

(Step 2: MPEP 1412.02(II.)(B)) The record of the prior proceedings 11/860,254 (US 8,648,867 B2) and 14/147,015 (US 9,189,828 B2) indicate:

a. In application 14/147,015, the Amendment of 03/10/2015 added to claim 1 (issued as claim 1 in patent 9,189,828) the following limitations in order to overcome the cited prior art (additions shown by underlining): a central processing unit to receive input data; main memory, operably coupled to the central processing unit via a bus, to store the input data received by the central processing unit; an accelerator, operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory, the accelerator comprising: at least one graphics processing unit to perform a sequence of computations on the at least a portion of the input data so as to generate output data, intermediate computations in the sequence of computations yielding intermediate results; and accelerator memory, operably coupled to the graphic processing unit, to store the results of the plurality of sequential computations; and a controller, operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory and to transfer at least a portion of the output data from the accelerator memory to the main memory during performance of the

sequence of computations by the at least one graphic processing unit. The accompanying remarks of 03/10/2015, on page 8, further indicate the following limitations overcome the cited prior art: "a controller … to transfer at least a portion of the output data from the accelerator memory to [a] main memory during performance of [a] sequence of computations by [a] graphic processing unit".

b. In application 14/147,015, the Amendment of 03/10/2015 added claim 12 (issued as claim 12 in patent 9,189,828) with the following limitations in order to overcome the cited prior art (additions shown by underlining): A method of performing a sequence of computations on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory, the method comprising: (A) performing, by the GPU, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data, intermediate computations in the sequence of computations yielding intermediate results; (B) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the input data from the main memory to the accelerator via the bus; and (C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus.

c. In application 11/860,254, the Amendment of 07/31/2013 added to claim 1 (issued as claim 1 in patent 8,648,867) the following limitations in order to overcome the cited prior art (additions shown by underlining): A computer system for performing a numerical simulation over a plurality of computational cycles including at least a first

computational cycle and a second computational cycle, the computer system comprising: a central processing unit; a main memory, operably coupled to the central processing unit, to store input data to be accessed by the central processing unit in performing the numerical simulation; a video system, operably coupled to the central processing unit, to drive a video monitor to display an indication of the numerical simulation in response to the computer system performing the numerical simulation; an accelerator, operably coupled to the central processing unit, to receive at least a portion of the input data from the central processing unit and to provide first output data generated during the first computational cycle to the central processing unit after a conclusion of the first computational cycle, the accelerator comprising: at least one graphics processing unit to generate second output data, during the second computational cycle, by performing at least one calculation on the first output data; and an accelerator memory, operably coupled to the at least one graphic processing unit, the accelerator memory comprising: a first partition, referenced by a first pointer, to store the first output data during the second computational cycle; and a second partition, referenced by a second pointer, to store the second output data generated during the second computational cycle; and an accelerator controller, operably coupled to the accelerator memory and the central processing unit, to transfer the at least the portion of the input data into the accelerator memory before the first computational cycle, to transfer the first output data from the accelerator memory to the main memory during the second computational cycle, to direct the second output data into the second partition during the second computational cycle, and to swap the first pointer and the second pointer at the conclusion of the second computational cycle such that the second output data becomes an input for a third computational cycle of

the plurality of computational cycles. The accompanying remarks of 07/31/2013, on pages 11-13, further indicate the following limitations overcome the cited prior art: "accelerator controller, operably coupled to the accelerator memory and the central processing unit, to transfer the at least the portion of the input data into the accelerator memory before the first computational cycle, to transfer the first output data from the accelerator memory to the main memory during the second computational cycle, to direct the second output data into the second partition at the end of the second computational cycle, and to swap the first pointer and the second pointer at the conclusion of the second computational cycle such that the second output data becomes an input for a third computational cycle of the plurality of computational cycles". The remarks argue claims 1 and 17 are similar in this regard.

d. In application 11/860,254, the Amendment of 07/31/2013 added to claim 10 (issued as claim 9 in patent 8,648,867) the following limitations in order to overcome the cited prior art (additions shown by underlining): An accelerator system for a computer system performing a numerical simulation, the accelerator system comprising: at least one graphics processing unit to generate output data by performing at least one computation during a first computational cycle of the numerical simulation; an accelerator memory, operably coupled to the at least one graphics processing unit, to store data used to perform the at least one computation; and an accelerator controller, operably coupled to the accelerator memory and the at least one graphics processing unit, to execute: (i) a computational stream controlling performance of the at least one computation by the at least one graphics processing unit; (ii) an output stream controlling transfer of the output data from the at least one graphics processing unit to the accelerator memory during the first computational cycle; and (iii) an input stream

controlling transfer of input data to the accelerator memory for use by the at least one graphics processing unit during a second computational cycle of the numerical simulation. The accompanying remarks of 07/31/2013, on page 14, further indicate the following limitations overcome the cited prior art: an accelerator controller that executes a computational stream to control performance of a first computation, an output stream to transfer data at the end of the first computational cycle, and an input stream to control the transfer of input data into the accelerator memory.

e. In application 11/860,254, the Amendment of 07/31/2013 added to claim 17 (issued as claim 16 in patent 8,648,867) the following limitations in order to overcome the cited prior art (additions shown by underlining): A method for performing a numerical simulation on input data in a computer system including a central processing unit and an accelerator, the method comprising: receiving, by an accelerator, first input data from the central processing unit; transferring, by an accelerator controller, the first input data into a first partition, referenced by first pointer, of an accelerator memory before a first computational cycle of the numerical simulation; performing, by at least one graphics processing unit during the first computational cycle, at least one calculation on the first portion of the input data as to generate first output data; storing, by the accelerator controller, the first output data into a second partition, referenced by a second pointer, of the accelerator memory; and swapping the first pointer with the second pointer at the end of the first computational cycle, such that the first output data becomes an input for a second computational cycle of the numerical simulation.

Subject matter is previously surrendered during the prosecution of the original application by amendment to define over the art OR reliance on an argument/statement made by applicant that a limitation of the claim(s) defines over the art. It is noted that a patent owner (reissue applicant) is bound by the amendment and/or argument that applicant relied upon to overcome an art rejection in the original application for the patent to be reissued, regardless of whether the Office adopted the argument in allowing the claims. Therefore, A) in the instant case the above cited limitations of the preceding paragraph are surrendered subject matter, and B) some of the broadening of the reissue claims, as noted above, is clearly in the area of the surrendered subject matter.

(Step 3: MPEP 1412.02(II.)(C)) It is noted that if surrendered subject matter has been entirely eliminated from a claim present in the reissue application, then a recapture rejection under 35 U.S.C. 251 is proper and must be made for that claim. Stated another way, if a claim limitation present in the original patent that was added to overcome a rejection or that was argued by applicant to distinguish over the prior art is entirely eliminated from a claim in the reissue application, then a recapture rejection under 35 U.S.C. 251 is proper and must be made for that claim. As illustrated in the above Step 1 and Step 2 section, the following surrendered limitations (in quotations) have been entirely removed, and not materially narrowed in other respects related to the surrendered subject matter:

a. From new independent claim 21: an accelerator, "operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory"; "intermediate computations in the sequence of computations yielding intermediate results"; and a controller, … to transfer at least a portion of the output data from the accelerator memory "to the main memory" (see above Step 1a and Step 2a).
b. From new independent claim 42: "on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator

operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory"; "the sequence of computations on a first portion of the input data so as to generate a first portion of the output data"; "intermediate computations in the sequence of computations yielding intermediate results"; "(B) in parallel with performing the sequence of computations by the GPU in (A)"; transferring a second portion of the input data "from the main memory to the accelerator via the bus"; and "(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus" (see above Step 1b and Step 2b).

c. From new independent claim 56: an accelerator, "operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory"; "intermediate computations in the sequence of computations yielding intermediate results;" accelerator memory, operably coupled to the graphic processing unit, "to store the results of the plurality of sequential computations"; and a controller, "operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of" the input data into the accelerator memory "and to transfer at least a portion of the" output data from the accelerator memory "to the main memory during performance of the sequence of computations by the" at least one graphic processing unit (see above Step 1c and Step 2a).

d. From new independent claim 62: "on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory"; "intermediate computations in the sequence of computations yielding intermediate results"; "(B) in parallel with performing the

sequence of computations by the GPU in (A); transferring a second portion of the input data from the main memory to the accelerator via the bus"; and "(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus" (see above Step 1d and Step 2b).

e. From new independent claim 68: "on a computer system comprising a … a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory"; "(A) performing, by the GPU, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data"; "intermediate computations in the sequence of computations yielding intermediate results"; "(B) in parallel with performing the sequence of computations by the GPU in (A) transferring a second portion of the input data from the main memory to the accelerator via the bus"; and "(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus" (see above Step 1e and Step 2b).

f. From new independent claim 79: an accelerator, "operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory"; at least one graphics processing unit "to perform a sequence of computations on the at least a portion of the input data so as to generate output data, intermediate computations in the sequence of computations yielding intermediate results"; accelerator memory, "operably coupled" to the at least one graphic processing unit, "to store the results of the plurality of sequential computations"; and a controller, "operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of" the input data into the accelerator memory "and to transfer at least a portion of the" output

data from the accelerator memory "to the main memory during performance of the sequence of computations by the at least one graphic processing unit" (see above Step 1f and Step 2a).

g. From new independent claim 90: "on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory"; and "via the bus" (see above Step 1g and Step 2b).

h. From new independent claim 95: an accelerator, "operably coupled to the central processing unit and the first memory via the bus, to receive at least a portion of the input data from the main memory"; accelerator memory, "operably coupled to the at least one graphic processing unit, to store the results of the plurality of sequential computations"; and a controller, "operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory and to transfer at least a portion of the output data from the accelerator memory to the main memory during performance of the sequence of computations by the at least one graphic processing unit" (see above Step 1h and Step 2a).

i. From new independent claim 101: A method for performing a numerical simulation "on input data" in a computer system "including a central processing unit and an accelerator", the method comprising: "receiving, by an accelerator, first input data from the central processing unit"; "transferring, by" an accelerator controller, "the first" input data "into a first partition, referenced by first pointer, of an accelerator memory before a first computational cycle of the numerical simulation"; "performing, by at least one graphics processing unit during the first computational cycle, at least one calculation on the first portion of the input data as to generate first output data"; "storing, by the accelerator controller"; "of the accelerator memory"; and

"swapping the first pointer with the second pointer at the end of the first computational cycle, such that the first output data becomes an input for a second computational cycle of the numerical simulation" (see above Step 1i and Steps 2c and 2e).

j. From new independent claim 108: at least one graphics processing unit "to generate output data by performing at least one computation during a first computational cycle of the numerical simulation"; an accelerator memory, "operably coupled to the at least one" graphics processing unit; and an accelerator controller, "operably coupled to the accelerator memory and the at least one graphics processing unit, to execute: (i) a computational stream" controlling performance of the at least one computation "by the at least one graphics processing unit"; "(ii) an output stream controlling transfer of the output" data "from" the at least one graphics processing unit to the accelerator memory during the first computational cycle; and "(iii) an input stream" controlling transfer of input data "to the accelerator memory" for use by the "at least one graphics processing unit" (see above Step 1j and Step 2d).

k. From new independent claim 112: a computer system including a central processing unit "and an accelerator", comprising: "receiving, by an accelerator, first input data" from the central processing unit; performing, by "at least one graphics processing unit"; storing, by "the accelerator controller", the first output data into a second partition, referenced by a second pointer, of "the accelerator memory" (see above Step 1k and Steps 2c and 2e).

Therefore improper recapture of broadened claimed subject matter surrendered in the application is clearly present in the instant reissue application. The respective dependent claims are similarly rejected based on their respective dependency from the independent claims and none of the claims contain the original combination of limitations presented.

Claim Rejections - 35 USC § 102 and § 103

In the event the determination of the status of the application as subject to AIA 35 U.S.C. 102 and 103 (or as subject to pre-AIA 35 U.S.C. 102 and 103) is incorrect, any correction of the statutory basis for the rejection will not be considered a new ground of rejection if the prior art relied upon, and the rationale supporting the rejection, would be the same under either status.

The following is a quotation of the appropriate paragraphs of pre-AIA 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

The following is a quotation of pre-AIA 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claim(s) 1-7, 10-16, 19-40, 42-43, 47-54, 56-72, 74-77, 79-83, 85-88, 90, 92-95, 97-100, and 117-123, 126-132, and 135-136 is/are rejected under pre-AIA 35 U.S.C. 102(e) as being anticipated by US 7,219,085 B2 *(herein **Buck**)*.

Claim 1

Buck discloses a computer system *(Buck: figure 1)*, comprising:

a central processing unit to receive input data *(Buck: figure 1, elements 120 and 160; figure 2, element 120; column 9, lines 33-36)*;

main memory, operably coupled to the central processing unit via a bus, to store the input data received by the central processing unit *(Buck: figure 1, elements 121, 130, 141, 152, 156)*;

an accelerator, operably coupled to the central processing unit and the main memory via the bus *(Buck: figure 1, element 190)*, to receive at least a portion of the input data from the main memory *(Buck: column 21, lines 54-64)*, the accelerator comprising:

at least one graphics processing unit to perform a sequence of computations on the at least a portion of the input data so as to generate output data *(Buck: figure 2, element 220; figure 5; figure 7; column 12, lines 16-26)*, intermediate computations in the sequence of computations yielding intermediate results *(Buck: figure 5, elements 595, temporary data, or 565, intermediate versions of W)*; and

accelerator memory, operably coupled to the at least one graphic processing unit, to store the results of the sequence of computations *(Buck: figures 2 and 5, element 230)*; and

a controller, operably coupled to the at least one graphics processing unit and the accelerator memory, to transfer the at least a portion of the input data into the accelerator memory and to transfer at least a portion of the output data from the accelerator memory to

the main memory during performance of the sequence of computations by the at least one graphic processing unit *(Buck: column 11, line 12 to column 12, line 26)*.

Claim 2

Buck discloses the computer system of claim 1, wherein the central processing unit is configured to receive the input data in response to a user interaction *(Buck: column 7, lines 51-55; column 23, lines 13-15)*.

Claim 3

Buck discloses the computer system of claim 1, wherein: the central processing unit is configured to receive the input data at a first rate; and the at least one graphics processing unit is configured to perform the sequence of computations at a second rate different than the first rate *(Buck: column 2, lines 55-67; column 9, lines 10-14; column 14, lines 40-44; CPU and GPU function at different rates)*.

Claim 4

Buck discloses the computer system of claim 1, wherein the main memory is configured to store a copy of the output data stored in the accelerator memory *(Buck: column 7, lines 39-51; figure 5, elements 580 and 585)*.

Claim 5

Buck discloses the computer system of claim 1, wherein an output of at least one computation in the sequence of computations represents an output of at least one neuron in an artificial neural network *(Buck: figure 7; column 22, lines 26-27; output of a neural network is an output layer of neurons)*.

Claim 6

Buck discloses the computer system of claim 1, wherein accelerator memory comprises: a first memory bank to store parameters common to all of the computations in the sequence of computations; and a second memory bank to store data specific to at least one computation in the sequence of computations *(Buck: figure 2, 230; figure 5, elements 530, 535, 595, 510; column 10, lines 36-63; see also column 20, lines 1-4)*.

Claim 7

Buck discloses the computer system of claim 1, wherein the controller is configured to transfer the output data from the accelerator memory to the main memory without transferring any of the intermediate results from the accelerator memory to the main memory so as to reduce data transfer via the bus *(Buck: figure 5, elements 595, temporary data, or 565, intermediate versions of W; figure 5, temporary data is never sent back; and column 11, lines 12-41, some embodiments do not sent back the intermediate versions of W)*.

Claim 10

Buck discloses the computer system of claim 1, wherein the controller is configured to control execution of the sequence of computations by the at least one graphics processing unit *(Buck: column 12, lines 16-26)*.

Claim 11

Buck discloses the computer system of claim 1, further comprising: at least one of a video camera, a microphone, or a cell recording electrode, operably coupled to the central processor unit, to acquire the

input data in real time *(Buck: column 7, line 51 to column 8, line 2; column 9, lines 33-36; column 23, lines 26-30)*.

Claim 12

Buck discloses a method of performing a sequence of computations on a computer system comprising a central processing unit (CPU), a main memory operably coupled to the central processing unit via a bus, an accelerator operably coupled to the CPU and the main memory via the bus, the accelerator comprising a graphics processing unit (GPU) and an accelerator memory *(Buck: figure 1, elements 120 and 160; figure 2, element 120; column 9, lines 33-36; figure 1, elements 121, 130, 141, 152, 156; figure 1, element 190; figure 2, elements 220 and 230)*, the method comprising:

(A) performing, by the GPU, the sequence of computations on a first portion of the input data so as to generate a first portion of the output data *(Buck: figure 2, element 220; figure 5; figure 7; column 12, lines 16-26)*, intermediate computations in the sequence of computations yielding intermediate results *(Buck: figure 5, elements 595, temporary data, or 565, intermediate versions of W)*;

(B) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the input data from the main memory to the accelerator via the bus *(Buck: column 11, line 12 to column 12, line 26)*; and

(C) in parallel with performing the sequence of computations by the GPU in (A), transferring a second portion of the output data from the accelerator memory to the main memory via the bus *(Buck: column 11, line 12 to column 12, line 26)*.

Claims 13-16 and 19-20

The limitations of claims 13-16 and 19-20 correspond to the limitations of claims 2-7 and 10-11 and as such are rejected in a corresponding manner.

Claim 21

Buck discloses an accelerator system *(Buck: figure 2, element 190)* comprising:

a graphics processing unit (GPU) to perform a sequence of computations on input data so as to generate output data *(Buck: figure 2, element 220; figure 5; figure 7; column 12, lines 16-26)*, an output of a computation in the sequence of computations representing an output of a neuron in an artificial neural network *(Buck: figure 7; column 22, lines 26-27; output of a neural network is an output layer of neurons)*;

a memory, operably coupled to the GPU, to store the input data and the output data *(Buck: figure 5, element 230)*; and

a controller, operably coupled to the GPU and the memory, to transfer the input data into the memory and to transfer at least a portion of the output data from the memory during performance of the sequence of computations by the GPU *(Buck: figure 5, inputs and outputs; column 11, line 59 to column 12, line 6)*.

Claim 22

Buck discloses the accelerator system of claim 21, wherein the GPU is operably coupled to a central processing unit (CPU) of a computer system *(Buck: figure 2)*.

Claim 23

Buck discloses the accelerator system of claim 21, wherein the GPU is mounted on an expansion card operably coupled to a motherboard of a computer system *(Buck: figure 2, element 210; column 9, lines 15-22, AGP and PCI Express busses are for expansion cards coupled to a motherboard)*.

Claim 24

Buck discloses the accelerator system of claim 21, further comprising: a central processing unit (CPU) *(Buck: figure 1, element 120)*, a system memory *(Buck: figure 1, element 130)*, and a non-volatile memory installed on a motherboard *(Buck: figure 1, element 133; column 6, lines 27-40, at least ISA bus indicates motherboard)*.

Claim 25

Buck discloses the accelerator system of claim 24, wherein the GPU is operably coupled to the CPU via a system bus on the motherboard *(Buck: figure 1, element 190 containing the GPU; figure 2, elements 190 and 210; column 9, lines 15-22, AGP and PCI Express busses are for expansion cards coupled to a motherboard)*.

Claim 26

Buck discloses the accelerator system of claim 25, wherein the system bus has a bandwidth of 3.2 GB/s to 15 GB/s *(Buck: column 9, lines 15-22, PCI Express bus has a bandwidth higher than 3.2GB/s)*.

Claim 27

Buck discloses the accelerator system of claim 25, wherein the controller is configured to transfer the input data and the at least a portion of the output data via the system bus *(Buck: figure 2, element 210)*.

Claim 28

Buck discloses the accelerator system of claim 25, further comprising: a video card, operably coupled to the motherboard via a local bus, to transfer graphical information to a display *(Buck: figure 1, elements 190, 121, and 191; claim language "local bus" is reference language "system bus")*.

Claim 29

Buck discloses the accelerator system of claim 28, wherein the local bus has a lower bandwidth than the system bus *(Buck: figure 1, element 121; column 6, lines 27-40, at least ISA bus; claim language "local bus" is reference language "system bus"; figure 2, element 210; column 9, lines 15-22; ISA bus has a lower bandwidth than a PCI Express bus)*.

Claim 30

Buck discloses the accelerator system of claim 21, wherein the memory comprises: a first memory partition to store parameters common to every computation in the sequence of computations *(Buck: figure 2, 230; figure 5, elements 530, 535, 595, 510; column 10, lines 36-63; see also column 20, lines 1-4)*.

Claim 31

Buck discloses the accelerator system of claim 30, wherein the memory further comprises: a second memory partition to store data specific to the computation in the sequence of computations *(Buck: figure 2, 230; figure 5, elements 530, 535, 595, 510; column 10, lines 36-63; see also column 20, lines 1-4)*.

Claim 32

Buck discloses the accelerator system of claim 31, wherein the second memory partition comprises: a first partition to hold external input data patterns *(Buck: figure 5, element 530; column 10, lines 61-62)*; a second partition to hold representations of internal variables *(Buck: column 15, lines 49-60)*; a third partition to hold an input of the computation in the sequence of computations *(Buck: column 10, lines 43-44)*; and a fourth partition to hold the output of the computation in the sequence of computations *(Buck: figure 5, element 510)*.

Claim 33

Buck discloses the accelerator system of claim 31, wherein the second memory partition is configured to accumulate the output of the computation in the sequence of computations during execution, by the GPU, of computational elements of the computation in the sequence of computations *(Buck: column 11, lines 50-52)*.

Claim 34

Buck discloses the accelerator system of claim 31, wherein the second memory partition is configured to store an output of a previous computation in the sequence of computations, the output of the previous computation being accessible to computational elements executed by the GPU during the computation in the sequence of computations *(Buck: figure 5, at least element 510; column 11, line 12 to column 12, line 15)*.

Claim 35

Buck discloses the accelerator system of claim 21, wherein the computation in the sequence of computations performed by the GPU comprises a plurality of computational elements *(Buck: column 21, lines 61-64)*.

Claim 36

Buck discloses the accelerator system of claim 35, wherein the plurality of computational elements represents a layer of neurons in an artificial neural network *(Buck: figure 7; column 21, lines 61-64; column 22, lines 26-27; output of a neural network is an output layer of neurons)*.

Claim 37

Buck discloses the accelerator system of claim 36, wherein all neurons in the layer of neurons are described by the same equation *(Buck: figure 7, same neuron type divided by layer; column 13, lines 33-55; and column 11, lines 33-36)*.

Claim 38

Buck discloses the accelerator system of claim 21, wherein the memory is configured to store intermediate results of the sequence of computations *(Buck: figure 5, elements 595, temporary data, or 565, intermediate versions of W)* and the controller is configured to transfer the output data from the memory to a main memory without transferring the intermediate results *(Buck: figure 5, temporary data is never sent back; and column 11, lines 12-41, some embodiments do not sent back the intermediate versions of W)*.

Claim 39

Buck discloses the accelerator system of claim 21, wherein the controller is configured to initiate transfer of at least a portion of the input data *(Buck: column 10, lines 36-43)* and to transfer the at least a portion of the output data in parallel *(figure 5, element 585; and column 6, lines 35-40, at least ISA bus is 8-bit wide indicating at least a portion of data is in parallel)*.

Claim 40

Buck discloses the accelerator system of claim 21, wherein the input data is acquired by at least one of a video camera, a microphone, or a cell recording electrode *(Buck: column 7, line 51 to column 8, line 2; column 9, lines 33-36; column 21, lines 54-64)*.

Claim 42

Buck discloses a method of performing a sequence of computations representing an artificial neural network *(Buck: column 21, lines 61-64)*, the method comprising:

performing, by a graphics processing unit (GPU), a first computation in the sequence of computations on first input data to generate first output data *(Buck: figure 2, element 220; figure 5; figure 7; column 12, lines 16-26)*, an output of a computation in the sequence of computations representing an output of a neuron in the artificial neural network *(Buck: figure 7; column 22, lines 26-27; output of a neural network is an output layer of neurons)*;

storing, in a memory operably coupled to the GPU, the first input data and the first output data *(Buck: figure 5, element 230)*; and

transferring second input data into the memory during performance of the sequence of computations by the GPU *(Buck: figure 5, inputs and outputs; column 11, line 59 to column 12, line 6)*.

Claim 43

Buck discloses the method of claim 42, wherein transferring the second input data comprises transferring the second input data via a bus operably coupled to a central processing unit (CPU) *(Buck: figure 2)*.

Claim 47

Buck discloses the method of claim 42, further comprising: storing, in a first memory partition of the memory, parameters common to all of the computations in the sequence of computations *(Buck: figure 2, 230; figure 5, elements 530, 535, 595, 510; column 10, lines 36-63; see also column 20, lines 1-4)*.

Claim 48

Buck discloses the method of claim 47, further comprising: storing, in a second memory partition of the memory, data specific to the first computation in the sequence of computations *(Buck: figure 2, 230; figure 5, elements 530, 535, 595, 510; column 10, lines 36-63; see also column 20, lines 1-4)*.

Claim 49

Buck discloses the method of claim 48, further comprising: storing, in the second memory partition, external input data patterns, representations of internal variables, an input of the computation in the sequence of computations, and the output of the computation in the sequence of computations *(Buck: figure 5, element 530; column 10, lines 61-62; column 15, lines 49-60; column 10, lines 43-44; figure 5, element 510)*.

Claim 50

Buck discloses the method of claim 42, wherein storing the first output data comprises: accumulating, in the memory, outputs of computational elements executed by the GPU in performing the first computation in the sequence of computations *(Buck: column 11, lines 50-52)*.

Claim 51

Buck discloses the method of claim 42, further comprising: storing, in the memory, an output of a previous computation in the sequence of computations; and accessing, by the GPU, the output of the previous computation during performance of the computation in the sequence of computations *(Buck: figure 5, at least element 510; column 11, line 12 to column 12, line 15)*.

Claim 52

Buck discloses the method of claim 42, wherein performing the first computation comprises executing a plurality of computational elements representing a layer of neurons in an artificial neural network *(Buck: figure 7; column 21, lines 61-64; column 22, lines 26-27; output of a neural network is an output layer of neurons)*.

Claim 53

Buck discloses the method of claim 52, wherein all neurons in the layer of neurons are described by the same equation *(Buck: figure 7, same neuron type divided by layer; column 13, lines 33-55; and column 11, lines 33-36)*.

Claim 54

Buck discloses the method of claim 42, further comprising: acquiring the input data with at least one of a video camera, a microphone, or a cell recording electrode *(Buck: column 7, line 51 to column 8, line 2; column 9, lines 33-36; column 21, lines 54-64)*.

Claim 56

Buck discloses an accelerator system *(Buck: figure 2, element 190)* comprising:

a graphics processing unit (GPU) to perform a sequence of computations on input data so as to generate output data *(Buck: figure 2, element 220; figure 5; figure 7; column 12, lines 16-26)*, a computation in the sequence of computations comprising a plurality of computational elements, the plurality of computational elements representing a layer of neurons in an artificial neural network *(Buck: column 21, lines 61-64)*, an output of the computation in the sequence of computations representing an output of a neuron in the artificial neural network *(Buck: figure 7; column 22, lines 26-27; output of a neural network is an output layer of neurons)*; and

a memory, operably coupled to the GPU, to store parameters common to all of the computations in the sequence of computations and data specific to the computation in the sequence of computations *(Buck: figure 2, 230; figure 5, elements 530, 535, 595, 510; column 10, lines 36-63; see also column 20, lines 1-4)*.

Claim 57

Buck discloses the accelerator system of claim 56, wherein all neurons in the layer of neurons are described by the same equation *(Buck: figure 7, same neuron type divided by layer; column 13, lines 33-55; and column 11, lines 33-36)*.

Claim 58

Buck discloses the accelerator system of claim 56, wherein the memory is configured to hold: external input data patterns *(Buck: figure 5, element 530; column 10, lines 61-62)*; representations of internal variables *(Buck: column 15, lines 49-60)*; an input to the computation in the sequence of computations *(Buck: column 10, lines 43-44)*; and the output of the computation in the sequence of computations *(Buck: figure 5, element 510)*.

Claim 59

Buck discloses the accelerator system of claim 56, wherein the memory is configured to accumulate the output of the computation in the sequence of computations during execution of the plurality of computational elements by the GPU *(Buck: column 11, lines 50-52)*.

Claim 60

Buck discloses the accelerator system of claim 56, wherein the memory is configured to store an output of a previous computation in the sequence of computations, the output of the previous computation being accessible to the plurality of computational elements during the computation in the sequence of computations *(Buck: figure 5, at least element 530; column 11, line 12 to column 12, line 15)*.

Claim 61

Buck discloses the accelerator system of claim 56, further comprising: a controller, operably coupled to the GPU and the memory, to transfer the input data into the memory and to transfer at least a portion of the output data from the memory during performance of the sequence of computations by the GPU *(Buck: column 11, line 12 to column 12, line 15)*.

Claim 62

Buck discloses a method comprising:

performing, with a graphics processing unit (GPU), a sequence of computations on input data to generate output data *(Buck: figure 2, element 220; figure 5; figure 7; column 12, lines 16-26)*, a computation in the sequence of computations comprising a plurality of computational elements, the plurality of computational elements representing a layer of neurons in an artificial neural network *(Buck:*

column 21, lines 61-64), an output of the computation in the sequence of computations representing an output of a neuron in the artificial neural network *(Buck: figure 7; column 22, lines 26-27; output of a neural network is an output layer of neurons)*;

storing, in a memory operably coupled to the GPU, parameters common to all of the computations in the sequence of computations, and storing, in the memory, data specific to the computation in the sequence of computations *(Buck: figure 2, 230; figure 5, elements 530, 535, 595, 510; column 10, lines 36-63; see also column 20, lines 1-4)*.

Claim 63-67

The limitations of claims 63-67 correspond to the limitations of claims 57-61 and as such are rejected in a corresponding manner.

Claim 68

Buck discloses a method of executing computations representing an artificial neural network *(Buck: column 21, lines 54-64)* on a computer system comprising at least one central processing unit (CPU) *(Buck: figure 2, element 120)*, a processing unit *(Buck: figure 2, element 220)*, a first memory partition *(Buck: figure 2, element 130)*, and a second memory partition *(Buck: figure 2, element 230)*, the method comprising:

executing, by the at least one CPU, a user interaction stream *(Buck: column 9, lines 35-36)*, the user interaction stream controlling transfer of inputs to the first memory partition and the second memory partition *(Buck: figures 2 and 5; column 10, lines 41-43)*;

executing, by the processing unit, a computational stream *(Buck: column 2, lines 24-28; column 8, lines 39-44)*, the computational stream controlling data exchange between the user interaction

stream and the computational stream during execution of the computations representing the artificial neural network *(Buck: figure 5, elements 220 and 545)*.

Claim 69

Buck discloses the method of claim 68, wherein the user interaction stream controls data exchange between the user interaction stream and the computational stream outside of execution of the computations representing the artificial neural network *(Buck: figure 2, element 130 separated by element 120)*.

Claim 70

Buck discloses the method of claim 68, wherein executing the user interaction stream comprises: controlling setting and editing of computational elements of the computations representing the artificial neural network *(Buck: figure 5, see at least 'P')*.

Claim 71

Buck discloses the method of claim 68, wherein executing the user interaction stream comprises: controlling setting and editing of parameters of the computations representing the artificial neural network *(Buck: figure 5, see at least 'W', 'L', 'ERRORS')*.

Claim 72

Buck discloses the method of claim 68, wherein executing the user interaction stream comprises: controlling setting and editing of parameters of the inputs to the artificial neural network *(Buck: figure 5, see at least 'X'' and 'T')*.

Claim 74

Buck discloses the method of claim 68, wherein executing the user interaction stream comprises: parsing elements to be used in the computations representing the artificial neural network *(Buck: column 2, lines 10-14; column 6, lines 10-16; column 10, lines 36-38; column 14, line 19 to column 15, line 18; programming and languages include parsing)*.

Claim 75

Buck discloses the method of claim 74, wherein the processing unit comprises a graphics processing unit (CPU) and executing the user interaction stream further comprises: converting the elements into GPU programs *(Buck: column 2, lines 10-14; column 6, lines 10-16; column 10, lines 36-38; column 14, line 19 to column 15, line 18; programming and languages include converting)*.

Claim 76

Buck discloses the method of claim 75, wherein executing the user interaction stream comprises: compiling the GPU programs *(Buck: column 2, lines 10-14; column 6, lines 10-16; column 10, lines 36-38; column 14, line 19 to column 15, line 18; DirectX includes compilation)*.

Claim 77

Buck discloses the method of claim 76, wherein executing the user interaction stream comprises: transferring the GPU programs to the second memory partition *(Buck: figure 5; and column 10, lines 36-41)*.

Claims 79-83 and 85-87

The limitations of claims 79-83 and 85-87 correspond to the limitations of claims 68-72 and 74-77 and as such are rejected in a corresponding manner.

Claim 88

Buck discloses the system of claim 87, wherein the user interaction stream transfers the GPU programs to a shader memory partition of the at least one memory *(Buck: figure 5; and column 10, lines 36-41)*.

Claims 90, 92-94, 95, and 97-100

The limitations of claims 90, 92-94, 95, and 97-100 correspond to the limitations of claims 1-7, 10-16, and 19-20 and as such are rejected in a corresponding manner.

Claims 117-123, 126-132, and 135-136

The limitations of claims 117-123, 126-132, and 135-136 correspond to the limitations of claims 1-7, 10-16, and 19-20 and as such are rejected in a corresponding manner.

Claim(s) 101-104, 106-109, and 111-112 is/are rejected under pre-AIA 35 U.S.C. 102(b) as being anticipated by US 2001/0010034 A1 *(herein **Burton**)*.

Claim 101

Burton discloses a method for performing a numerical simulation over a plurality of computational cycles including at least a first computational cycle and a second computational cycle *(Burton: figure 7; paragraph 0075)*, the method comprising:

storing, in a first partition of a memory, first output data during the second computational cycle, the first output data generated during the first computational cycle *(Burton: figure 7; paragraph 0075;*

the current cycle, which corresponds to the claimed second cycle, uses data that it is storing in memory area A and which was previously generated in a first cycle) and being referenced by a first pointer *(Burton: figure 7, element 38)*;

performing, by a processing unit operably coupled to the memory, at least one calculation on the first output data during the second computational cycle, the at least one calculation yielding second output data *(Burton: figure 7, elements 32-36; paragraph 0075; the current cycle generates new output data)*;

storing, in a second partition of the memory, the second output data, the second output data being referenced by a second pointer *(Burton: figure 7, elements 36 and 38; paragraph 0075; writing output to memory area B)*; and

swapping the first pointer and the second pointer at the conclusion of the second computational cycle such that the second output data becomes an input for a third computational cycle of the plurality of computational cycles *(Burton: figure 7, element 38; the new output data can now be used for another future cycle, which corresponds to the claimed third cycle)*.

Claim 102

Burton discloses the method of claim 101, further comprising: transferring input data into the memory from another memory coupled to a central processing unit (CPU) before the first computational cycle *(Burton: figures 6-7; paragraphs 0074-0075; input data for the claim labeled "first cycle" coming from an even earlier cycle of Burton's loop)*.

Claim 103

Burton discloses the method of claim 101, further comprising: providing, to a central processing unit (CPU), first output data generated during the first computational cycle after a conclusion of the first

computational cycle *(Burton: figures 6-7; paragraphs 0074-0075; showing the CPU is the controlling/executing everything and is thus provided with the output and adjusting/swapping the pointers in all cycles)*.

Claim 104

Burton discloses the method of claim 101, further comprising: transferring the first output data from the memory to another memory during the second computational cycle *(Burton: figure 7, element 38)*.

Claim 106

Burton discloses the method of claim 101, further comprising: dictating, to the processing unit, an order of execution of instructions *(Burton: paragraph 0075, "compile time scheduling")*.

Claim 107

Burton discloses the method of claim 101, wherein performing the at least one calculation comprises receiving, by the processing unit, at least the portion of the input data and a series of instructions and executing, by the processing unit, the instructions on the at least the portion of the input data *(Burton: paragraph 0074-0075, 0078, and 0042; executing compiled instructions from memory describing the simulation and the hardware, and including data of the simulation)*.

Claim 108

Burton discloses a method of performing a numerical simulation *(Burton: figure 7; paragraph 0075)*, the method comprising:

performing, by a processing unit, at least one computation during a first computational cycle of the numerical simulation *(Burton: figure 7; paragraph 0075; first iteration of the loop)*;

storing, by a memory operably coupled to the processing unit, data used to perform the at least one computation *(Burton: figure 7; paragraph 0075; memory area A storing inputs to the first iteration of the loop)*;

controlling performance of the at least one computation by the processing unit *(Burton: figure 7; paragraph 0075; executing generating output during the first iteration of the loop)*;

controlling transfer of the output data from the processing unit to the memory during the first computational cycle *(Burton: figure 7; paragraph 0075; writing output to memory area B during the first iteration of the loop)*; and

controlling transfer of input data to the memory for use by the processing unit during a second computational cycle of the numerical simulation *(Burton: figure 7; paragraph 0075; swapping pointers and then reading from memory area A the old output, which is now new input, during the second iteration of the loop)*.

Claim 109

Burton discloses the method of claim 108, further comprising: dictating an order of execution of instructions to the processing unit *(Burton: paragraph 0075, "compile time scheduling")*.

Claim 111

Burton discloses the method of claim 108, further comprising: perform successive computational cycles of the numerical simulation by feeding output data generated by the processing unit from a previous computational cycle and input data for a next computational cycle into the processing unit *(Burton: figure 7; paragraph 0075; output of the preceding cycle is input for the next cycle)*.

Claim 112

Burton discloses a system for performing a numerical simulation on input data *(Burton: figure 7; paragraph 0075)*, the system comprising:

a central processing unit (CPU) to provide first input data *(Burton: figure 6, element 22)*;

a memory, operably coupled to the CPU, to store the first input data in a first partition, referenced by first pointer, before a first computational cycle of the numerical simulation *(Burton: figure 6, element 24)*;

a processing unit, operably coupled to the memory, to perform, during a first computational cycle, at least one calculation on the first portion of the input data so as to generate first output data *(Burton: figure 6, CPU; figure 7; paragraph 0075; first iteration of the loop)*; and

a controller, operably coupled to the processing unit and the memory, to store the first output data in a second partition of the memory *(Burton: figure 7; paragraph 0075; memory area B)*, the second partition referenced by a second pointer, and to swap the first pointer with the second pointer at the end of the first computational cycle, such that the first output data becomes an input for a second computational cycle of the numerical simulation *(Burton: figure 7, element 38; paragraph 0075)*.

Claim 8-9, 17-18, 44-46, 96, 124-125, and 133-134 is/are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over US 7,219,085 B2 *(herein **Buck**)* in view of US 7,525,547 B1 *(herein **Diard**)*.

Claim 8

To the extent ***Buck*** does not explicitly state the computer system of claim 1, wherein the controller is configured to transfer at least a portion of the output data from the accelerator memory to the main memory after the at least one graphics processing unit has begun to perform another sequence of computations, ***Diard*** demonstrates that it was known at the time of invention to transfer output from an accelerator memory to the main memory while the GPU is performing other operations *(Diard: figure*

1, transfer arrows; column 5, lines 1-3; column 13, lines 4-25). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the CPU and GPU system of ***Buck*** with output transferred from the accelerator memory to the main memory after the GPU has started other computations as suggested by ***Diard***'s teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art; and both ***Buck*** and ***Diard*** are directed to CPU and GPU processing.

Claim 9

To the extent ***Buck*** does not explicitly state the computer system of claim 8, wherein the controller is configured to initiate transfer of the at least a portion of the input data and to transfer the at least a portion of the output data in parallel with performance of at least one computation in the other sequence of computations by the at least one graphics processing unit. ***Diard*** demonstrates that it was known at the time of invention to transfer output from an accelerator memory to the main memory while the GPU is performing other operations *(Diard: figure 1, transfer arrows; column 5, lines 1-3; column 13, lines 4-25)*. ***Diard*** further demonstrates simultaneous transfer of input data and output data by an accelerator controller *(Diard: column 6, lines 10-26; column 13, lines 4-25)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the CPU and GPU system of ***Buck*** with initiating transfer of input data and transferring output data in parallel with a GPU's computation in another sequence of computations as suggested by ***Diard***'s teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art; and both ***Buck*** and ***Diard*** are directed to CPU and GPU processing.

Claims 17-18

The limitations of claims 17-18 correspond to the limitations of claims 8-9 and as such are rejected in a corresponding manner.

Claim 44

Buck discloses the method of claim 42, further comprising: transferring the first output data from the memory to another memory during a second computation in the sequence of computations. ***Diard*** demonstrates that it was known at the time of invention to transfer output from a memory to another memory during a computation *(Diard: figure 1, transfer arrows; column 5, lines 1-3; column 13, lines 4-25)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the CPU and GPU system of ***Buck*** with output transferred from the accelerator memory to the main memory after the GPU has started other computations as suggested by ***Diard***'s teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art; and both ***Buck*** and ***Diard*** are directed to CPU and GPU processing.

Claim 45

Buck discloses the method of claim 44, further comprising: storing intermediate results of the sequence of computations in the memory, and wherein transferring the first output data from the memory to the other memory occurs without transferring the intermediate results of the sequence of computations *(Buck: figure 5, elements 595, temporary data, or 565, intermediate versions of W; figure 5, temporary*

data is never sent back; and column 11, lines 12-41, some embodiments do not sent back the intermediate versions of W).

Claim 46

Buck discloses the method of claim 44, wherein transferring the second input data and transferring the first output data occurs in parallel *(Buck: figure 5, element 585; and column 6, lines 35-40, at least ISA bus is 8-bit wide indicating at least a portion of data is in parallel)*. Also, ***Diard*** demonstrates that it was known at the time of invention to transfer output from an accelerator memory to the main memory while the GPU is performing other operations *(Diard: figure 1, transfer arrows; column 5, lines 1-3; column 13, lines 4-25)*. ***Diard*** further demonstrates simultaneous transfer of input data and output data by an accelerator controller *(Diard: column 6, lines 10-26; column 13, lines 4-25)*.

Claim 96

To the extent ***Buck*** does not explicitly state the system of claim 95, wherein the first memory partition and the second memory partition are logical partitions, ***Diard*** demonstrates that it was known at the time of invention for various memory to be shared or logically partitioned between graphics units and CPUs *(Diard: column 4, lines 5-11)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the memory of ***Buck*** as logical partitions as suggested by ***Diard***'s teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art; and both ***Buck*** and ***Diard*** are directed to CPU and GPU processing.

Claims 124-125 and 133-134

The limitations of claims 124-125 and 133-134 correspond to the limitations of claims 8-9 and 17-18 and as such are rejected in a corresponding manner.

Claims 41, 55, and 91 is/are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over US 7,219,085 B2 *(herein **Buck**)* in view of US 7,119,810 B2 *(herein **Sumanaweera**)*.

Claim 41

To the extent ***Buck*** does not explicitly state the accelerator system of claim 21, wherein the input data is loaded from disk. ***Sumanaweera*** demonstrates that it was known at the time of invention to load GPUs from disk *(Sumanaweera: column 1, lines 10-19)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement ***Buck*** with input data loaded from disk as suggested by ***Sumanaweera***'s teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art.

Claims 55 and 91

The limitations of claims 55 and 91 correspond to the limitations of claim 41 and as such are rejected in a corresponding manner.

Claims 73, 78, 84, and 89 is/are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over US 7,219,085 B2 *(herein **Buck**)* in view of US 2006/0129506 A1 *(herein **Edelman**)*.

Claim 73

To the extent, ***Buck*** does not explicitly state the method of claim 68, wherein executing the user interaction stream comprises: specifying an output to be saved to disk and/or displayed on a screen. ***Edelman*** demonstrates that it was known at the time of invention to save computations of an output stream to a disk *(Edelman: paragraphs 0056, and 0079)* and for the output stream to be related to an artificial or simulated neural system *(Edelman: paragraphs 0056, and 0079)*. Further, ***Buck*** demonstrates CPU controlled outputs and disk memory disk *(Buck: figures 1 and 2; column 3, lines 46-51; column 9, lines 15-37)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the artificial neural network of ***Buck*** with saving output to disk as suggested by ***Edelman***'s teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art.

Claim 78

To the extent ***Buck*** does not state the method of claim 68, further comprising: executing, by the at least one CPU, a data output stream, the data output stream controlling transfer of outputs of the computations representing the artificial neural network to disk. ***Edelman*** demonstrates that it was known at the time of invention to save computations of an output stream to a disk *(Edelman: paragraphs 0056, and 0079)* and for the output stream to be related to an artificial or simulated neural system *(Edelman: paragraphs 0056, and 0079)*. Further, ***Buck*** demonstrates CPU controlled outputs and disk memory disk *(Buck: figures 1 and 2; column 3, lines 46-51; column 9, lines 15-37)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the artificial neural network of ***Buck*** with saving output to disk as suggested by ***Edelman***'s teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation

of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art.

Claims 84 and 89

The limitations of claims 84 and 89 correspond to the limitations of claims 73 and 78 and as such are rejected in a corresponding manner.

Claim 105 is/are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over US 2001/0010034 A1 *(herein **Burton**)* in view of US Patent 5,142,665 *(herein **Bigus**)*.

Claim 105

To the extent ***Burton*** does not explicitly state the method of claim 101, further comprising: displaying, with a video system, an indication of the numerical simulation. ***Bigus*** demonstrates that it was known at the time of invention to display simulation results *(Bigus: column 3, lines 32-34; column 6, lines 35-41; column 10, lines 29-33)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the simulations of ***Burton*** with displaying an indication of the simulation as suggested by ***Bigus'*** teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art.

Claims 110 and 113-116 is/are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over US 2001/0010034 A1 *(herein **Burton**)* in view of US 7,219,085 B2 *(herein **Buck**)*.

Claim 110

To the extent ***Burton*** does not explicitly state the method of claim 108, further comprising: transferring output data from the memory to a main memory while the processing unit performs the at least one computation. ***Buck*** demonstrates that it was known at the time of invention to transfer output data to another memory, for example out to a main memory of a CPU, while the processing unit performs a computation *(Buck: column 11, line 12 to column 12, line 26; also figures 1 and 2)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the processing unit of ***Burton*** as a CPU making use of a GPU with transferring output data to main memory during a computation as suggested by ***Buck's*** teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art.

Claim 113

To the extent ***Burton*** does not explicitly state the system of claim 112, wherein the processing unit comprises a graphics processing unit. ***Buck*** demonstrates that it was known at the time of invention to off load processing to a graphics processing unit *(Buck: figure 5)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the processing unit of ***Burton*** as a CPU making use of a GPU as suggested by ***Buck's*** teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art.

Claim 114

To the extent ***Burton*** does not explicitly state the system of claim 112, wherein the controller is configured to send instructions for performing the at least one calculation to the processing unit. ***Buck*** demonstrates that it was known at the time of invention for a controller to send instructions for performing a calculation to a processing unit *(Buck: column 10, lines 34 to column 12, line 26)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the processing unit of ***Burton*** as a GPU with a controller to send instructions for performing a calculation as suggested by ***Buck's*** teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art.

Claim 115

To the extent ***Burton*** does not explicitly state the system of claim 112, wherein the accelerator controller is further configured to transfer the first output data to another memory during the second computational cycle. ***Buck*** demonstrates that it was known at the time of invention to transfer output data to another memory, for example out to a CPU, during a second computational cycle *(Buck: column 11, line 12 to column 12, line 26)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the processing unit of ***Burton*** as a CPU making use of a GPU with a controller transferring output data to another memory during a second computation cycle as suggested by ***Buck's*** teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art.

Claim 116

To the extent ***Burton*** does not explicitly state the system of claim 112, wherein the memory further comprises: a third partition to store internal variables; and a fourth partition to store data used as input at a particular computation cycle of the numerical simulation. ***Buck*** demonstrates that it was known at the time of invention for a memory to have a partition to store internal variables and a partition to store data used as input at a computational cycle *(Buck: figure 5, element 230)*. It would have been obvious to one of ordinary skill in the art at the time of invention to implement the processing unit of ***Burton*** as a CPU making use of a GPU with a memory partitioned with partitions for internal variables, data used as inputs to a computational cycle, as well as input data and output data as suggested by ***Buck's*** teachings. This implementation would have been obvious because one of ordinary skill in the art would have found: the implementation of is a substitution and application of one known element and technique for another yielding a predictable result using an acceptable piece of prior art.

Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to William H. Wood whose telephone number is (571)272-3736 and fax number is (571)273-3736. The examiner can normally be reached on Monday-Thursday 7am-4pm; Friday 7am-11am. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alexander Kosowski can be reached on (571)272-3744. The Central Reexamination Unit's (CRU) fax number is (571)273-9900.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR systems, see https://ppair.uspto.gov/epatent/portal/home. For questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/William H. Wood/
Primary Examiner, CRU 3992

Conferees:
/RSD/

/ALEXANDER J KOSOWSKI/
Supervisory Patent Examiner, Art Unit 3992