

Demonstratives of Patent Owner Autonomous Devices, LLC

USPTO Patent Trial and Appeal Board
Oral Hearing: July 31, 2025

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

1,474



US011663474B1

(12) **United States Patent**
Cosic

(10) **Patent No.:** **US 11,663,474 B1**
(45) **Date of Patent:** ***May 30, 2023**

(54) **ARTIFICIALLY INTELLIGENT SYSTEMS, DEVICES, AND METHODS FOR LEARNING AND/OR USING A DEVICE'S CIRCUMSTANCES FOR AUTONOMOUS DEVICE OPERATION**

(71) Applicant: **Jasmin Cosic**, Miami, FL (US)

(72) Inventor: **Jasmin Cosic**, Miami, FL (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.: **17/561,976**

(22) Filed: **Dec. 26, 2021**

Related U.S. Application Data

(63) Continuation of application No. 16/540,972, filed on Aug. 14, 2019, now Pat. No. 11,238,344, which is a continuation of application No. 15/340,991, filed on Nov. 2, 2016, now Pat. No. 10,452,974.

(51) Int. Cl. **G06N 3/08** (2023.01)
G06N 5/022 (2023.01)

(52) U.S. Cl. **G06N 3/08** (2013.01); **G06N 5/022** (2013.01)
CPC

(58) **Field of Classification Search**
CPC **G06N 5/022; G06N 3/08**
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,370,707 A 1/1983 Phillips et al.
4,730,315 A 3/1988 Saito et al.

4,860,203 A 8/1989 Corrigan et al.
5,560,011 A 9/1996 Uyama
5,602,982 A 2/1997 Judd et al.
5,874,834 A 2/1999 New
6,026,234 A 2/2000 Hanson et al.
6,088,731 A 7/2000 Kiraly et al.
6,106,292 A 8/2000 Ackermann et al.
6,126,330 A 10/2000 Knight
6,344,558 B1 11/2001 Angel et al.
6,408,262 B1 6/2002 Leerberg et al.
6,643,842 B2 11/2003 Angel et al.
6,728,689 B1 4/2004 Drissi et al.
6,735,632 B1 5/2004 Kiraly et al.
6,754,631 B1 6/2004 Din
6,801,912 B2 10/2004 Moskowitz et al.
6,842,877 B2 1/2005 Roberts et al.
6,915,105 B2 7/2005 Masuda
6,973,446 B2 12/2005 Mamitsuka et al.
6,996,758 B1 2/2006 Herron et al.

(Continued)

OTHER PUBLICATIONS

Tracing (software), retrieved from <URL: <http://wikipedia.com>> on Jan. 10, 2014, 3 pages.

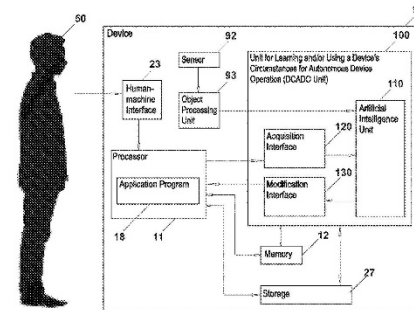
(Continued)

Primary Examiner Santiago Garcia
(74) Attorney, Agent, or Firm — Imperium Patent Works LLP

(57) **ABSTRACT**

Aspects of the disclosure generally relate to computing enabled devices and/or systems, and may be generally directed to devices, systems, methods, and/or applications for learning a device's operation in various circumstances, storing this knowledge in a knowledgebase (i.e. neural network, graph, sequences, etc.), and enabling autonomous operation of the device.

84 Claims, 40 Drawing Sheets



Overview of Issues

<i>Inter Partes Review Grounds</i>	
Ground 1A	Obviousness of Claims 56-57 based on Buibas-Sinyavskiy
Ground 2C	Obviousness of Claims 56-57 based on Grotmol-Buibas
<i>Motion to Amend Issues</i>	
Issue 1	Written Description Support for Substitute Claims
Issue 2	Obviousness of Limitations [85.11]-[85.12] based on Grotmol-Zhu
Issue 3	Obviousness of Limitations [85.13]-[85.14] based on Grotmol-Zhu (and Grotmol-Zhu-Buibas)

● **Claims 56 and 57**

- Ground 1A: Buibas-Sinyavskiy Does Not Disclose or Suggest:
 - Inputs Being Correlated with Another One or More Instruction Sets for Operating a Third Device (Claims 56/57)
 - At Least a Portion of the Instruction Sets for Operating the Third Device is Learned in Another Learning Process that Includes Operating the Third Device by Another User (Claim 56)
- Ground 2C: Grotmol-Buibas Does Not Disclose or Suggest:
 - Inputs Being Correlated with Another One or More Instruction Sets for Operating a Third Device (Claims 56/57)
 - At Least a Portion of the Instruction Sets for Operating the Third Device is Learned in Another Learning Process that Includes Operating the Third Device by Another User (Claim 56)

Claim 45

45. A first device comprising:
a knowledgebase that includes one or more inputs for inputting at least a portion of a circumstance representation, wherein the one or more inputs are correlated with one or more instruction sets for operating a second device;
one or more sensors;
one or more processors; and
one or more non-transitory machine readable media storing machine readable code that, when executed by the one or more processors, causes the one or more processors to perform at least:

generating a circumstance representation, wherein the generated circumstance representation represents a circumstance detected at least in part by the one or more sensors;
determining the one or more instruction sets for operating the second device at least by: inputting at least a portion of the generated circumstance representation into the one or more inputs, and using a correlation between the one or more inputs and the one or more instruction sets for operating the second device; and
at least in response to the determining, executing the one or more instruction sets for operating the second device, wherein the first device autonomously performs one or more operations defined by the one or more instruction sets for operating the second device.

Ex. 1001 ('474 pat.) at 176:20-46

Claim 46

46. The first device of claim **45**, wherein the one or more inputs for inputting the at least the portion of the circumstance representation include one or more inputs for inputting at least a portion of one or more object representations, and wherein the generated circumstance representation includes one or more object representations, and wherein the first device is a first vehicle, and wherein the second device is a second vehicle.

Ex. 1001 ('474 pat.) at 176:47-54

Claims 56 and 57

56. The first device of claim 46, wherein the one or more inputs are further correlated with another one or more instruction sets for operating a third device, and wherein at least a portion of the one or more instruction sets for operating the second device or at least an information related to the one or more instruction sets for operating the second device is learned in a learning process that includes operating the second device at least partially by a user, and wherein at least a portion of the another one or more instruction sets for operating the third device or at least an information related to the another one or more instruction sets for operating the third device is learned in another learning process that includes operating the third device at least partially by another user.

57. The first device of claim 46, wherein the one or more inputs are further correlated with another one or more instruction sets for operating a third device.

Ex. 1001 ('474 pat.) at 177:58-178:7

- Claims 56 and 57
- **Ground 1A: Buibas-Sinyavskiy Does Not Disclose or Suggest:**
 - **Inputs Being Correlated with Another One or More Instruction Sets for Operating a Third Device (Claims 56/57)**
 - At Least a Portion of the Instruction Sets for Operating the Third Device is Learned in Another Learning Process that Includes Operating the Third Device by Another User (Claim 56)
- Ground 2C: Grotmol-Buibas Does Not Disclose or Suggest:
 - Inputs Being Correlated with Another One or More Instruction Sets for Operating a Third Device (Claims 56/57)
 - At Least a Portion of the Instruction Sets for Operating the Third Device is Learned in Another Learning Process that Includes Operating the Third Device by Another User (Claim 56)

Disputed Limitation

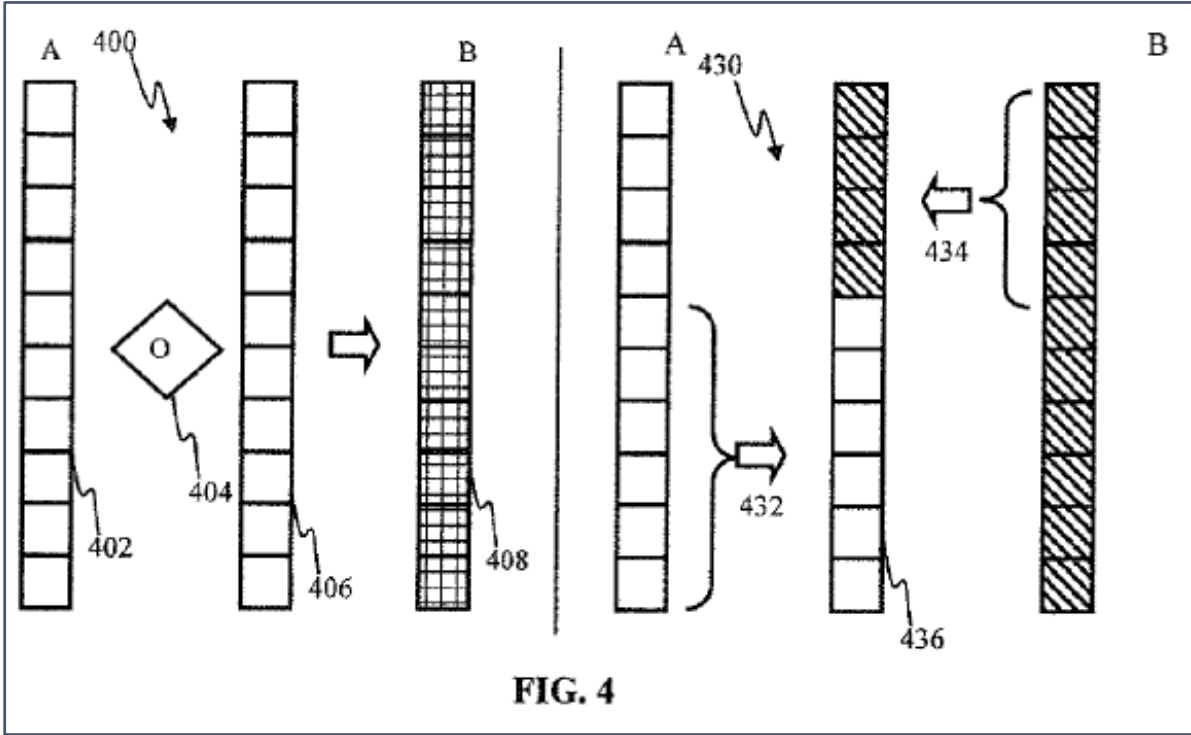
57. The first device of claim **46**, wherein the one or more inputs are further correlated with another one or more instruction sets for operating a third device.

Ex. 1001 ('474 pat.) at 178:5-7

Dr. Saber's Declaration

63. In the Petition, Petitioner cites its reasoning regarding claim element 45.1, Petition, 41-42, but in my opinion that claim element does not include further correlating the one or more inputs with *another* instruction set for operating *a third device*, as required by claim 56. Claim 45 only mentions instruction sets for operating a second device. Ex. 1001, Claim 45 (reciting for element 45.1: “a knowledgebase that includes one or more inputs for inputting at least a portion of a circumstance representation, wherein the *one or more inputs are correlated with one or more instruction sets for operating a second device*”).

Buibas



Ex. 1005 (Buibas) at FIG. 4

[0028] FIG. 4 is a functional block diagram illustrating a network merge in accordance with one or more implementations.

73. Petitioner's further citations to Buibas ¶[0189] (generally discussing combining brain images), and ¶[0225] (generally discussing merging brain images) also never mention instruction sets and further correlating the same one or more inputs with another instruction set for operating a third device as required by claim 56. In my opinion, Buibas ¶[0189] and ¶[0225] discuss merging brain images very generally without even a mention of *additional another* instruction set(s) for operating *a third device*, the *same one or more inputs*, and how they are *further* correlated as required by the claim. In addition, Buibas makes clear - in its other sections as described below- what its brain image merges actually involve and how that happens:

Ex. 2001 (Saber Decl.) at ¶73

Buibas

[0123] At step 810, the target network configuration is generated. In one or more implementations, the target configuration may comprise network weights downloaded into the target robotic device. In some implementations, the target configuration may comprise network weights vector stored within the cloud server and available for subsequent downloads to one or more robotic devices (e.g., 610 of FIG. 6).

Ex. 1005 (Buibas) at [0123]

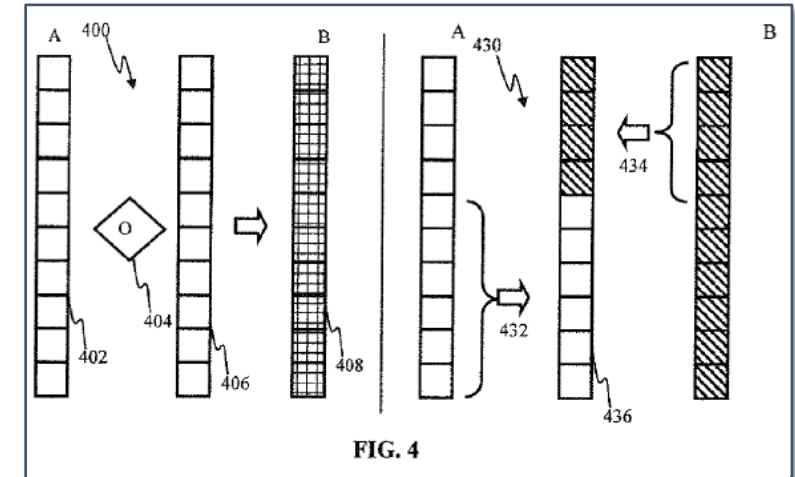
606. An authenticated user connecting through a user interface device 1202 connects to the cloud server, and identifies a brain image 1404 to be loaded onto the computerized neuromorphic apparatus 1150. The computerized neuromorphic apparatus downloads the selected brain image, and set its synaptic weights appropriately. In one or more implementations, the brain image 1404 may comprise the network weights so that the download effectuates the weight configuration of the computerized neuromorphic apparatus network. The user may access personal content and/or public content (e.g. shared/purchasable content). Alternatively, the exemplary networking architecture illustrated in FIG. 12A may be used and the cloud server 606 connects to the neuromorphic apparatus 1150 using the user interface device 1202 as a bridge.

Buibas

Network Merges

[0094] In some implementations, the network lifecycle may comprise network weight updates comprising two or more sources, also referred herein to as “merges”.

[0095] In one or more implementations, such as the merge 400 illustrated in FIG. 4, the ‘offspring’ network configuration (e.g., the weight vector W) 408 may be obtained using two parent weight vectors 402, 406 combined via an operation 404.



Ex. 1005 at FIG. 4

Buibas

[0096] In some implementations, the network merge may comprise non-interleaved combination as illustrated by the merges 430 in FIG. 4, where the parents A provides the portion 432 and the parent B provides the portion 434 to the off-spring 436. In some implementations (not shown), the portion 432 may comprises the same number of weights as the portion 434. The non-interleaved network merge, such as illustrated in FIG. 4, may be used, for example, to combine capabilities of two (or more) parents or to add a capability to one of the parent network.

Ex. 1005 (Buibas) at [0096]

[0097] Various operations 404 of FIG. 4 may be used to produce contributions by one or more parent. In one or more implementations, the off-spring weight vector may be determined by combining the traits from parent A and parent B are combined as follows:

$$W^{off} = (W^A \text{ OR } W^B) \quad (\text{Eqn. 3})$$

[0098] In one or more implementations, the off-spring weight vector may be determined by determining traits in the parent B that are not in the parent A and adding these to the parent A as follows:

$$W^{off} = (W^A \text{ AND } W^B). \quad (\text{Eqn. 4})$$

$W^{off} = (W^A \text{ OR } W^B)$ (Eqn. 3)

$W^{off} = (W^A \text{ AND } W^B)$. (Eqn. 4)

Ex. 1005 (Buibas) at [0097]-[0098]

8 Q. Can you explain to me how equation 3 of
9 Buibas works?

10 A. So my recollection is that this is an
11 ORing function where a decision is made between a
12 weight or weight vector of parents A and parent --
13 parents A and B. And that weight vector or subset
14 of weights is then what constitutes the weight
15 vector for the offspring.

Ex. 2005 (Janet Tr.) at 8:8-15

20 Q. And parent A corresponds to the variable
 21 that is labeled as WA, and parent B is the thing
 22 that corresponds to WB in equation 3; right?
 1 A. When you say "the thing" --
 2 Q. The variable or parameter.
 3 A. Again, I just want to clarify. You're
 4 not ORing non-logic level values. What you're
 5 doing is you're using the non-logic level values,
 6 i.e., these weights, the synoptic weights, to
 7 ascertain whether there was utility if the neural
 8 network even trained on it. And then that defines
 9 your logic level zero or one.

Ex. 2005 (Janet Tr.) at 11:20-12:9

$$W^{off} = (W^A \text{ OR } W^B) \quad (\text{Eqn. 3})$$

$$W^{off} = (W^A \text{ AND } W^B). \quad (\text{Eqn. 4})$$

Ex. 1005 (Buibas) at [0097]-[0098]

14 Q. And in equation 4, there are two input
 15 terms, WA corresponding to parent A, and WB
 16 corresponding to parent B. And they are put into
 17 the AND operation, and the result of that is
 18 called WOFF; is that correct?
 19 A. For offspring, correct.
 20 Q. Why are there two variations of this
 21 type of an operation to combine traits from two
 22 parent -- two parents?

Ex. 2005 at 13:14-22 (center),
 14:1-10 (right)

1 A. Well, this just shows that there are
 2 lots of criteria that can be used to determine
 3 whether to prune, graft, augment a neural network
 4 with the knowledge that was gained by another
 5 neural network. Buibas tried to simplify it in
 6 this equation -- or these two equations.
 7 But, again, it's a very well-known
 8 process, dating back decades, where if there's
 9 value, which can usually be determined by looking
 10 at the weights or by operating a network without
 11 the weight to see what the difference might be,
 12 adding the neurons and associated weights and
 13 seeing what the difference might be.
 14 So this is the most simplified way of
 15 describing what would otherwise be a very broad
 16 spectrum of criteria to determine whether
 17 something of weight or a set of weights has value
 18 within a neural network. And then if it does, how
 19 to fold it in.

1 Q. And all of the terms in equations 3 and
 2 4, both the input terms and the resulting output
 3 of those equations, are all weight vectors;
 4 correct?

5 A. The weight vectors are represented in
 6 these equations, but as I mentioned before, you're
 7 not logically comparing non-logic-level values.
 8 There's a criteria -- again, if you look at the
 9 paragraph that talks about whether there are
 10 traits in parent B that are not in parent A.

Ex. 2005 (Janet Tr.) at 17:1-10

$$W^{eff} = (W^A \text{ OR } W^B) \quad (\text{Eqn. 3})$$

$$W^{eff} = (W^A \text{ AND } W^B). \quad (\text{Eqn. 4})$$

Ex. 1005 (Buibas) at [0097]-[0098]

19 Q. Are there any other terms in equation 3
 20 other than WA and WB that go into the input of the
 21 OR function?

22 A. Those are the only terms that are
 1 represented. I don't know how to make this any
 2 more clear, but a person of ordinary skill would
 3 look at this and for a practical implementation
 4 would understand that we're talking about traits,
 5 we're talking about weights that would enhance or
 6 play a role in those traits.

7 And if those traits exist, then it might
 8 sit in a logic level one category. If they don't
 9 exist and they might otherwise have value, then
 10 they might sit in a logic level zero.

Ex. 2005 at 18:19-19:10
 Autonomous Ex. 2009 p. 18 of 90

Dr. Saber's Declaration

68. First, a network merge is a term Buibas uses for a merge of weight vectors of the neural networks. The entire section of Buibas titled “*Network Merges*” (Ex. 1005, ¶[0094-0098]) *specifically describes merging of weight vectors*. Further, Buibas specifically states that “merges” are “network weight updates comprising two or more sources.” Ex. 1005, ¶[0094]. Even Further, Petitioner’s own citations of Buibas clearly show that a network merge is a merge of weight vectors of the neural networks (see Ex. 1005, ¶[0028] stating “FIG. 4 is a functional block diagram illustrating a *network merge* in accordance with one or more implementations” and *showing merging of weight vectors*; see also Ex. 1005, FIG. 4 (showing an example of merging of weight vectors and using the resulting merged weight vector to update weights of a neural network)).

Dr. Saber's Declaration

69. Second, in my opinion, Petitioner's contention of combining capabilities by a network merge (i.e. merging weight vectors as Buibas specifically defines a "network merge" as discussed above) is not, and indeed does not even mention, further correlating the same one or more inputs with another instruction set, let alone an instruction set for operating a third device as required by claim 56. Merging two weight vectors as Buibas specifically discloses in its "Network Merges" involves combining weights of weight vectors and does not involve any instruction sets, much less *further* correlating *the same* one or more inputs with *another* instruction set *for operating a third device* as a part of the merge process, and as required by claim 56. Further correlating instruction sets would require additional training or retraining to properly incorporate the added instructions into the network.

PO Sur-Reply

Apparently recognizing that Buibas' merges just involve weights (Reply, 7, Petitioner admitting "it is true that the discussion of FIG. 4 centers on merging network weights"), Petitioner provides various cites and arguments regarding Buibas' brain images. Reply, 7-10. But, *none of those cites disclose or suggest instruction sets at all, much less correlation of inputs with instruction sets* for operating any device. The statement in the Reply that "Buibas specifically explains

[0203] In an exemplary implementation, the brain image serves as a spatial mapping (in a simulated space) of the neurons and synapses of an artificial neural network. The brain image is expressed as a sparse matrix (allows for empty space, growth, and connectivity mapping). Other approaches may be used consistent with the disclosure as well.

Ex. 1005 (Buibas) at [0203]

Petitioner's Reply

Moreover, regardless of how the particular type of merge employed, a POSITA would have understood from Buibas' explanation that the offspring retains the full capabilities of the parent networks that the offspring incorporates sufficient aspects of the parents that allow it to perform the same functions and thus produce the same or equivalent outputs for a given input as the parents (e.g., output instructions sets for operating a second device and additional instructions sets for operating a third device, where those outputs are part of the respective capabilities of the parents). TESLA-1022, ¶17. The incorporated aspects of the parent networks

PO Sur-Reply

and is unsupported. **First**, capabilities is such a generality that if the claimed “instruction sets” would be allowed to be included in the definition of “capabilities,” anything could be included without a meaningful limit, which is clearly unreasonable. **Second**, “capabilities” is actually a different thing than the claimed “instruction sets” by any definition or interpretation that a POSITA would use. For example, an instruction set may be code that instructs a device to turn right such as `Device.turnRight()`, whereas a capability does not include such code. **Third**, merging Buibas’ brain images updates or combines capabilities without involving inputs or instruction sets since Buibas’ brain images only include weights. In other words, updating or combining capabilities is due to the updating or merging weights of brain images, not inputs or instruction sets. **Fourth**, the Reply states that



Conclusory expert testimony does not qualify as substantial evidence. See, e.g., *MobileMedia Ideas LLC v. Apple Inc.*, 780 F.3d 1159, 1172 (Fed. Cir. 2015) (“Conclusory statements by an expert . . . are insufficient to sustain a jury’s verdict.”); *Regents of Univ. of Minn. v. AGA Med.*

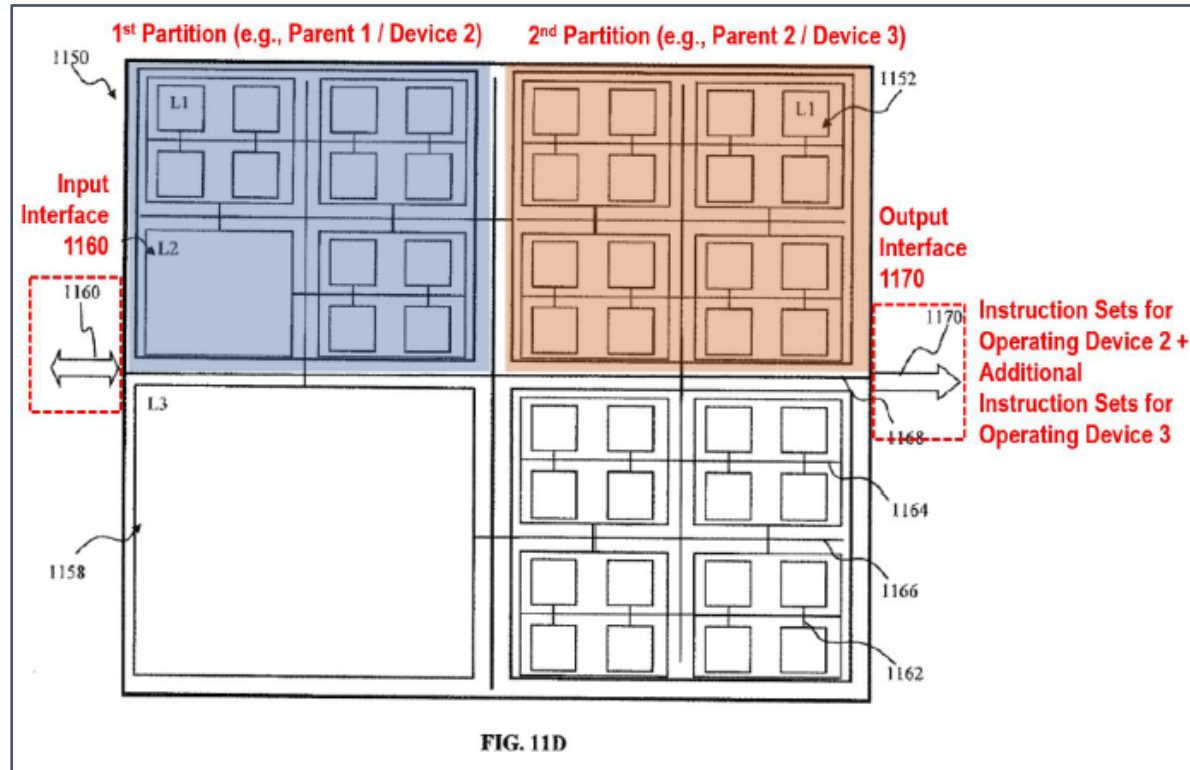
Corp., 717 F.3d 929, 941 (Fed. Cir. 2013) (quoting *Sitrick v. Dreamworks, LLC*, 516 F.3d 993, 1001 (Fed. Cir. 2008)) (“Conclusory expert assertions cannot raise triable issues of material fact”); *Koito Mfg. Co. v. Turn-Key-Tech, LLC*, 381 F.3d 1142, 1152 (Fed. Cir. 2004) (“General and conclusory testimony . . . does not suffice as substantial evidence of invalidity.”); *Sea Robin Pipeline Co. v. FERC*, 795 F.2d 182, 188 (D.C. Cir. 1986) (“[I]nordinate faith in the conclusory assertions of an expert . . . cannot satisfy the requirement . . . [of] substantial evidence.”).

TQ Delta, LLC v. Cisco Sys., Inc., 942 F.3d 1252, 1358-59 (Fed. Cir. 2019)

Buibas

[0102] Various exemplary implementations of the transformation operations 404, 505 are illustrated with respect to FIGS. 5B-5C. The merges 510, 520 illustrated in FIG. 5B comprise interleaved segments of weights from parent A and parent B (the segments 512, 522 from the parent A and the segments 514, 524 from the parent B, respectively, in FIG. 5B) to produce the off-spring weight vectors 516, 526, respectively. While the combination 510 comprises equal contributions from each parent, the combination 520 may comprise non-equal contributions, with the parent 522 providing a larger portion of the weights to the off-spring 526. Although the parent A, the parent B and the off-spring are shown as separate entities in FIG. 5B-5C for clarity, it will be appreciated by those skilled in the arts that such may not always be the case. By way of illustration, in some implementations, the offspring may comprise either (or both) of the parents A or B. In such implementation, a computerized device may be used to assist the merge store intermediate off-spring weight configuration during the merge, and to update one (or both) of the parents.

Buibas



Ex. 1005 (Buibas) at FIG. 11D
(annotated by Petitioner)

PO Sur-Reply

But Buibas does not say anything about one partition (annotated by Petitioner in blue) producing an instruction set for one device (annotated by Petitioner as Device 2) and another partition (annotated by Petitioner in orange) producing an instruction set for another device (annotated by Petitioner as Device 3). **Petitioner's arguments and annotations are unsupported in Buibas and are speculative, conclusory, and should be rejected.** Similarly, the Reply states that, in Buibas, the

Buibas

[0179] The training application 1204 also includes options management of the computerized neuromorphic apparatus 1150. Connectivity options for both the neuromorphic apparatus and the user interface device 1202 may be set from the training application. Link options between the apparatus 1150 and the device 1202 may also be configured. The training menu may be used to load active “brain images” (e.g., synaptic weight data configurations or profiles) onto the neuromorphic apparatus. In some cases, a user may load a brain image onto the apparatus 1150 that is insufficient to control all of the features of the robotic apparatus (e.g. no support for certain sensors). If these features are not controlled by other firmware/software on the robotic device, they may be rendered inactive (or otherwise inoperable). In other implemen-

Ex. 1005 (Buibas) at [0179]

[0144] One particular implementation of the computerized neuromorphic processing system, for operating a computerized spiking network (and implementing the exemplary inverse STDP context connection adjustment methodology described supra), is illustrated in FIG. 11A. The computerized system 1100 of FIG. 11A comprises an input interface 1110, such as for example an image sensor, a computerized spiking retina, an audio array, a touch-sensitive input device, etc. The input interface 1110 is coupled to the processing block (e.g., a single or multi-processor block) via the input communication interface 1114. The system 1100 further comprises a random access memory (RAM) 1108, configured to store neuronal states and connection parameters (e.g., weights 526 in FIG. 5), and to facilitate synaptic updates. In

Dr. Saber's Declaration

77. Further, additional instruction sets cannot simply be added to existing instruction sets that are already correlated with existing same inputs in a knowledgebase (i.e. neural network) without appropriately retraining the network. As of the critical date (see my discussion above at Sections III and V), additional instruction sets could only be added to existing instruction sets that are already correlated with existing inputs by appropriately retraining the knowledgebase with training sets that include the additional instructions sets. This is so because, during retraining, proper weights for correlations/connections are calculated/adjusted through backpropagation process (see Ex. 1001, 18:64-67) for specific instruction sets associated with specific same inputs in each

training set. Backpropagation, generally speaking, typically involves an iterative process of training a neural network where a training input is provided to a given network and the corresponding calculated output is computed via feed forward operations through the network. The error between the calculated and expected output is then backpropagated through the network using algorithms such as gradient descent in order to properly update the weights throughout the layers. This process is repeated multiple times with various inputs using an appropriately chosen training dataset until convergence is achieved. The optimized network is then tested and further optimized using a validation dataset and then further tested using a test dataset to measure/quantify its performance.

Dr. Saber's Declaration

80. In my opinion, *none of Petitioner's contentions, Petitioner's citations of Buibas, Buibas's figures, and Buibas's text assert, show, or describe a "merged" neural network where instruction sets of two neural networks are merged and properly integrated into the network.* Such merging of instruction sets of two neural networks would require appropriate technical explanation of how the additional instruction sets are correlated/connected with inputs via correlations/connections and/or intermediate neurons and how those correlations/connections are correctly weighted for proper statistical predictions

of those additional instructions sets, which requires retraining the knowledgebase (i.e. neural network) with training sets that include the additional instructions sets as discussed above. Buibas does not show or describe merging instruction sets of two neural networks or the underlying retraining process(es) that would be required to appropriately and seamlessly merge additional instructions sets. Instead, what Buibas actually shows and describes is the merging of weights in weight vectors of two neural networks that does not involve instruction sets.

Petitioner's Reply

5. Merging Weight Vectors by Itself Would Still Produce a Brain Image Having One or More Inputs Correlated with the Claimed Instruction Sets for Operating Second and Third Devices

Even if one accepts *arguendo* that Buibas' merge operation is limited to merging only network **weights** as PO contends (which is not correct—*see supra*, §II.A.2), Buibas would still render claims 56 and 57 obvious. TESLA-1022, ¶28.

Petitioner's Reply

analyzed the issue or Buibas's disclosure. Although weights and instruction sets are not the same, they are closely intertwined, particularly in the context of Buibas' neural networks that are configured to output instruction sets (e.g., motor commands) for operating a device. TESLA-1022, ¶28; TESLA-1005, [0153],

Dr. Saber's Declaration

82. In my opinion, a weight is a fundamentally different concept than the claimed instruction set. Weights are typically statistical “value[s]” or “numeric attribute[s],” Ex. 1001, 104:10-13, that can be “properties of a connection[],” Ex. 1005, ¶[0077]. Conversely, instruction sets are commands for operating a device or its actuators. Ex. 1001, 152:13-29. To illustrate the point, a clear fundamental difference between a weight and an instruction set of operating a device is that a weight is a number such as 0.238 that signifies a weight of an associated connection between two neurons, whereas an instruction set is typically a command such as “Device1.moveForward(27)... for moving a Device1 forward 27 units.” Ex. 1001, 142:53-57.

21 Q What about an instruction set? How
22 would you define an instruction set?

1 A In the context of Buibas, which is
2 reference 1005, the instruction set would be
3 the commands that would be given to motors and
4 various other components.

Ex. 2003 (Janet Tr.) at 48:22-49:4

4 A Let's unpack that for a second.

5 First of all, you can unpack -- you
6 can pull a weight vector, but that weight
7 vector is not going to tell you much about
8 that neural network. It's just going to be a
9 block of values.

Ex. 2003 at 80:4-9

18 Q Okay. And how would you or two-way
19 vectors together? How would that work?

6 If I were to do a graft,
7 for example, I would use -- and this was done
8 for my Ph.D., my dissertation -- I would pull
9 a portion of that neural network. That
10 portion would include, let's just say a neuron
11 and synaptic weights, biases, activation
12 functions, topology, and I would fold that
13 into another neural network, and I would get a
14 different means of correlating input values to
15 output, in this case, in the context of '474,
16 instruction sets.

Ex. 2003 (Janet Tr.) at 78:18-79:16

18 Q So you could extract a portion of a
19 neural network and fold that into another
20 neural network and -- I'm sorry, can you
21 explain that process again?

5 First of all, you can unpack -- you
6 can pull a weight vector, but that weight
7 vector is not going to tell you much about
8 that neural network. It's just going to be a
9 block of values.

Ex. 2003 at 79:18-80:9

3 And so if I get an array of values,
4 that's not enough. I'd need to know the
5 topology, the activation functions, feedback,
6 biases, et cetera.

7 Q To arrive at the instruction set,
8 right?

9 A Well, to contribute to arriving at
10 the one or more instruction sets.

Ex. 2003 at 84:10

Network Merges

[0094] In some implementations, the network lifecycle may comprise network weight updates comprising two or more sources, also referred herein to as “merges”.

[0097] Various operations 404 of FIG. 4 may be used to produce contributions by one or more parent. In one or more implementations, the off-spring weight vector may be determined by combining the traits from parent A and parent B as combined as follows:

$$W^{off} = (W^A \text{ OR } W^B) \quad (\text{Eqn. 3})$$

[0098] In one or more implementations, the off-spring weight vector may be determined by determining traits in the parent B that are not in the parent A and adding these to the parent A as follows:

$$W^{off} = (W^A \text{ AND } W^B). \quad (\text{Eqn. 4})$$

Ex. 1005 (Buibas) at [0094], [0097]-[0098]

[0179] The training application 1204 also includes options management of the computerized neuromorphic apparatus 1150. Connectivity options for both the neuromorphic apparatus and the user interface device 1202 may be set from the training application. Link options between the apparatus 1150 and the device 1202 may also be configured. The training menu may be used to load active “brain images” (e.g., synaptic weight data configurations or profiles) onto the neuromorphic apparatus. In some cases, a user may load a brain image onto the apparatus 1150 that is insufficient to control all of the features of the robotic apparatus (e.g. no support for certain sensors). If these features are not controlled by other firmware/software on the robotic device, they may be rendered inactive (or otherwise inoperable). In other implemen-

Ex. 1005 at [0179]

- Claims 56 and 57
- **Ground 1A: Buibas-Sinyavskiy Does Not Disclose or Suggest:**
 - Inputs Being Correlated with Another One or More Instruction Sets for Operating a Third Device (Claims 56/57)
 - **At Least a Portion of the Instruction Sets for Operating the Third Device is Learned in Another Learning Process that Includes Operating the Third Device by Another User (Claim 56)**
- Ground 2C: Grotmol-Buibas Does Not Disclose or Suggest:
 - Inputs Being Correlated with Another One or More Instruction Sets for Operating a Third Device (Claims 56/57)
 - At Least a Portion of the Instruction Sets for Operating the Third Device is Learned in Another Learning Process that Includes Operating the Third Device by Another User (Claim 56)

Disputed Limitation

56. The first device of claim 46, wherein the one or more inputs are further correlated with another one or more instruction sets for operating a third device, and wherein at least a portion of the one or more instruction sets for operating the second device or at least an information related to the one or more instruction sets for operating the second device is learned in a learning process that includes operating the second device at least partially by a user, and wherein at least a portion of the another one or more instruction sets for operating the third device or at least an information related to the another one or more instruction sets for operating the third device is learned in another learning process that includes operating the third device at least partially by another user.

Ex. 1001 ('474 pat.) 177:58-178:4

Petitioner's Reply

perform different tasks/behaviors. Brain images configured to perform the various tasks/behaviors learned on different devices by training routines carried out by different users operating their different robots can then be shared with other users, and users can then select and incorporate different ones of the learned tasks/behaviors into a merged brain image can be loaded on a single device. *See,*

Buibas

Merges” section). Petitioner ignores that Buibas’ brain image is not a full neural network with inputs, instruction sets, and weights, but instead is a group of weights (even if these weights may be from different sources) that are loaded into a neural network of a neuromorphic apparatus per Buibas’ explicit disclosure *See id.*, ¶ [0179] (“[t]he training menu [that] may be used to load active ‘brain images’ (e.g., synaptic *weight* data configurations or profiles) onto the neuromorphic apparatus.”). And a group (*i.e.*, vector) of weights does not include instruction sets, much less instruction sets for operating different devices learned in different learning processes involving different users.

PO Sur-Reply [26] at 17

[0179] The training application 1204 also includes options management of the computerized neuromorphic apparatus 1150. Connectivity options for both the neuromorphic apparatus and the user interface device 1202 may be set from the training application. Link options between the apparatus 1150 and the device 1202 may also be configured. The training menu may be used to load active “brain images” (e.g., synaptic weight data configurations or profiles) onto the neuromorphic apparatus. In some cases, a user may load a brain image onto the apparatus 1150 that is insufficient to control all of the features of the robotic apparatus (e.g. no support for certain sensors). If these features are not controlled by other firmware/software on the robotic device, they may be rendered inactive (or otherwise inoperable). In other implemen-

Ex. 1005 (Buibas) at [0179]

PO Sur-Reply

Graphics, Inc., 800 F.3d 1375, 1378 (Fed. Cir. 2015). Additionally, Buibas' separate training routines on different devices produce trained weights of brain images on different devices, not the claimed instruction sets for operating different devices since a brain image is not a neural network, includes only weights, and does not include instruction sets. Ex. 1005, ¶¶[0028], [0094]-[0095], [0097]-[0098], [0183]. Thus, Buibas' separate training routines on different devices produce something different (trained weights) than the claimed instruction sets for operating different devices.

Petitioner's Reply

with no specific explanation of its meaning. TESLA-1001, 103:10-19. In a related IPR involving similar subject matter and claims that likewise referred to multiple learning processes, PO's declarant, Dr. Saber, acknowledged that different learning processes may involve different users, different devices, and operating device(s) at different times (although not all are required). TESLA-1032, 38:18-39:2, 49:19-51:12, 52:11-55:21. All of those factors are nonetheless present in Buibas, whose platform allows different users to independently carry out different training routines on different devices at different times. These factors strongly show that the instruction sets learned on a parent third device in Buibas are learned in another learning process separate from the learning process on the second device. TESLA-1022, ¶33.

Dr. Saber's Declaration

91. Further, as I discussed above, any additional instruction sets (especially instruction sets learned in an additional or different learning process involving an additional user operating an additional device) *cannot be simply integrated with the existing instruction sets that are already correlated with the existing inputs without appropriately retraining the network via processes such as backpropagation in order to update the weights of the newly added connections to allow for the proper integration of the added instruction sets.*

Such retraining of the knowledgebase with additional instruction sets learned in an additional learning process involving an additional user operating an additional device is a complex process that is not straightforward or unusually simple. Further evidence of this is the fact that none of the prior art references mention such retraining.

- Claims 56 and 57
- Ground 1A: Buibas-Sinyavskiy Does Not Disclose or Suggest:
 - Inputs Being Correlated with Another One or More Instruction Sets for Operating a Third Device (Claims 56/57)
 - At Least a Portion of the Instruction Sets for Operating the Third Device is Learned in Another Learning Process that Includes Operating the Third Device by Another User (Claim 56)
- **Ground 2C: Grotmol-Buibas Does Not Disclose or Suggest:**
 - **Inputs Being Correlated with Another One or More Instruction Sets for Operating a Third Device (Claims 56/57)**
 - At Least a Portion of the Instruction Sets for Operating the Third Device is Learned in Another Learning Process that Includes Operating the Third Device by Another User (Claim 56)

Disputed Limitation

57. The first device of claim **46**, wherein the one or more inputs are further correlated with another one or more instruction sets for operating a third device.

Ex. 1001 ('474 pat.) at 178:5-7

Dr. Saber's Declaration

105. Petitioner's citations to its own arguments in claims 45.1, 47, and 51, Petition, 81, are wrong because claims 45, 47, and 51 recite "one or more inputs are correlated with one or more instruction sets for operating *a second device*" and do not recite the actual limitation of claim 56: "wherein the one or more inputs are further correlated with another one or more instruction sets for operating a third device." Claim 56 requires that the *same* one or more inputs be *further* correlated with an *additional another instruction set* for operating *an additional third device*, which is beyond what claims 45.1, 47, and 51 recite and what Petitioner's arguments for those claims address.

Dr. Saber's Declaration

instruction set for operating an *additional third device*. In other words, all of the instruction sets for operating the first robot where they were learned are still the same instruction sets for operating the same first robot (not a third robot) when they are loaded into a second robot in Grotmol. Since Grotmol discloses learning on one robot and loading this knowledge to a second robot without mentioning an instruction set for operating a third robot, Grotmol does not disclose or suggest that the same inputs are further correlated with another instruction for operating a third device, as claimed in claim 56. The seamless integration of additional or newly acquired instruction sets with existing instructions sets requires high levels to retraining to ensure that the entire network is properly optimized. This is not trivial but rather require skilled expertise to do so properly.

Petition

To the extent it is argued or determined that Grotmol does not expressly disclose that the inputs to an adaptive controller on a first device are correlated with control outputs (*instructions sets*) learned on at least two other devices (i.e., a *second device* and a *third device*), these features nonetheless would have been obvious when further considering the teachings of Buibas. TESLA-1003, ¶160. Buibas, for example, describes techniques for sharing and merging behaviors learned on multiple different robots into a single controller. TESLA-1005, ¶¶[0070], [0090]-[0100], [0103], [0225]. It would have been obvious to implement Grotmol in accordance with Buibas' suggestion sharing and merging behaviors learned on multiple different robots into a single controller for each of the reasons described above in the combination overview (*supra*, §IV.E.1). TESLA-1003, ¶160.

Buibas

[0090] FIG. 3 illustrates various exemplary implementations of robotic brain (network) updates. The network updates illustrated in FIG. 3 may comprise for instance a complete weight update 300, where the existing trained weights vector W 302 is replaced with a different weight vector 304. Such implementation may be used for example, when operating several identical robotic devices and the network weight configuration of a single 'trained' apparatus is downloaded into other (untrained) devices. The update implementation may also be used when (improved) training versions of network are distributed by one user (and/or a vendor) to other user with (or without) a cost associated therewith.

Ex. 1005 (Buibas) at [0090]

[0103] The interleaved network merge may be used in some implementations to generate new weight vector for a network configured to be use by off-spring configured to implement (a subset) of parent functionality. By way of illustration, two or more trash collector robotic apparatus, employed in a home or an office building, and comprising similar hardware (e.g., sensors and actuators) but trained under different conditions and, hence, comprising different sets of network weights, may utilize interleaved merge to produce an off-spring exhibiting combined set of capabilities.

Ex. 1005 at [0103]

[0183] Referring now to FIG. 14, a functional block diagram of an exemplary implementation of a cloud-based training system 1400 according to the disclosure is shown. A computerized neuromorphic apparatus 1150 uses on-board or accessible network connectivity to connect to a cloud server 606. An authenticated user connecting through a user interface device 1202 connects to the cloud server, and identifies a brain image 1404 to be loaded onto the computerized neuromorphic apparatus 1150. The computerized neuromorphic apparatus downloads the selected brain image, and set its synaptic weights appropriately. In one or more implementations, the brain image 1404 may comprise the network weights so that the download effectuates the weight configuration of the computerized neuromorphic apparatus network. The user may access personal content and/or public content (e.g. shared/purchasable content). Alternatively, the exemplary networking architecture illustrated in FIG. 12A may be used and the cloud server 606 connects to the neuromorphic apparatus 1150 using the user interface device 1202 as a bridge.

PO Sur-Reply

A POSITA would have understood from these disclosures that Grotmol's controller can first be trained on a third device to learn instruction sets for operating a third device; the trained configuration of the controller can be shared with and loaded on a second device where the trained configuration is updated through additional learning of instruction sets for operating the second device (e.g., in the override/correct modes); and the updated trained configuration can be shared with and loaded on a first device. TESLA-1022, ¶37. As Dr. Janet explains, this

Pet. Reply [19] at 24

configuration can be shared with and loaded on a first device.” Reply, 24. This is **conclusory**. Nothing like this convoluted scheme is disclosed or suggested in Grotmol. In fact, Grotmol does not even mention instruction sets for operating a third device, let alone inputs that are further correlated with instruction sets for operating a third device. Furthermore, Grotmol does disclose that a “trained configuration may be loaded to one or more other robots in order to provide learned behaviors,” but that has *nothing* to do with the claim limitation in claims 56/57: “the one or more inputs are further correlated with another one or more instruction sets for operating a third device.”

PO Sur-Reply [26] at 22

PO Sur-Reply

teachings of *Buibas*.”) (emphasis added). That is, Petitioner has made an improper new argument in its Reply that **changes its obviousness theory** for this ground—now Petitioner is arguing that the limitation would have been obvious in view of Grotmol alone, with no mention of Buibas on pages 22-24 of the Reply (the portion of the Reply addressing this limitation in Ground 2C). Reply, 24 (“As Dr. Janet explains,

Petition

To the extent it is argued or determined that Grotmol does not expressly disclose that the inputs to an adaptive controller on a first device are correlated with control outputs (*instructions sets*) learned on at least two other devices (i.e., a *second device* and a *third device*), these features nonetheless would have been obvious when further considering the teachings of Buibas. TESLA-1003, ¶160.

Pet. [2] at 81

Petitioner's Reply

and loaded on a first device. TESLA-1022, ¶37. As Dr. Janet explains, this predictable situation would have been expected and obvious to a POSITA reviewing Grotmol's disclosure, as it would allow multiple devices to gain the benefit of Grotmol's capabilities for sharing and updating training configurations. *Id.* The predictable result would be a trained controller on the first device whose inputs are correlated both with instruction sets for operating the second device (based on training that occurred on the second device) and with instruction sets for operating the third device (based on training that occurred on the third device). *Id.*

Pet. Reply [19] at 24

- Claims 56 and 57
- Ground 1A: Buibas-Sinyavskiy Does Not Disclose or Suggest:
 - Inputs Being Correlated with Another One or More Instruction Sets for Operating a Third Device (Claims 56/57)
 - At Least a Portion of the Instruction Sets for Operating the Third Device is Learned in Another Learning Process that Includes Operating the Third Device by Another User (Claim 56)
- **Ground 2C: Grotmol-Buibas Does Not Disclose or Suggest:**
 - Inputs Being Correlated with Another One or More Instruction Sets for Operating a Third Device (Claims 56/57)
 - **At Least a Portion of the Instruction Sets for Operating the Third Device is Learned in Another Learning Process that Includes Operating the Third Device by Another User (Claim 56)**

Disputed Limitation

56. The first device of claim 46, wherein the one or more inputs are further correlated with another one or more instruction sets for operating a third device, and wherein at least a portion of the one or more instruction sets for operating the second device or at least an information related to the one or more instruction sets for operating the second device is learned in a learning process that includes operating the second device at least partially by a user, and wherein at least a portion of the another one or more instruction sets for operating the third device or at least an information related to the another one or more instruction sets for operating the third device is learned in another learning process that includes operating the third device at least partially by another user.

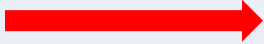
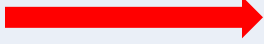
Ex. 1001 ('474 pat.) at 177:58-178:4

PO Sur-Reply

The Reply states regarding this limitation that Patent Owner’s “arguments on this limitation mirror the same flawed arguments that it earlier advanced in Ground 1A” and “lack merit and fail to refute the Petition’s showing of obviousness for each of the reasons addressed above in §§II and III.A.” Reply, 25. But it is Petitioner that is wrong about such arguments, for the reasons discussed above in Sections II-IV.

● **Proposed Substitute Claims**

- The Substitute Claims Have Adequate Written Description Support
- Limitations [85.11]-[85.12]: Grotmol-Zhu Does Not Disclose or Suggest the Specifically Claimed Multiple Object Representation Data Structures
- Limitations [85.13]-[85.14]: Grotmol-Zhu-Buibas Does Not Disclose or Suggest Determining One or More Instruction Sets for a Second and Third Device by Inputting the Specifically Claimed Multiple Object Representation Data Structures into Input Neurons and Using a Correlation Between the Input Neurons and Instruction Sets

Original Claim	Contingency	Substitute Claim
Claim 45		Claim 85
Claim 46		Claim 86
Claim 47		Claim 87
Claim 51		Claim 88
Claim 56		Claim 89
		Claim 92
Claim 57		Claim 90
		Claim 93
Claim 60		Claim 91



Substitute Claim 85

Claim 85 (Proposed substitute for claim 45, if claim 45 is found unpatentable):

[85.pre] A first device comprising:

[85.1] a knowledgebase that includes one or more inputs for inputting at least a portion of a circumstance representation, wherein the one or more inputs are correlated with one or more instruction sets for operating a second device;

[85.2] one or more sensors;

[85.3] one or more processors; and

[85.4] one or more non-transitory machine readable media storing machine readable code that, when executed by the one or more processors, causes the one or more processors to perform at least:

[85.5] generating a circumstance representation, wherein the generated circumstance representation represents a circumstance detected at least in part by the one or more sensors;

[85.6] determining the one or more instruction sets for operating the second device at least by: inputting at least a portion of the generated circumstance representation into the one or more inputs, and using a correlation between the one or more inputs and the one or more instruction sets for operating the second device; and

[85.7] at least in response to the determining, executing the one or more instruction sets for operating the second device, wherein the first device autonomously performs one or more operations defined by the one or more instruction sets for operating the second device;

[85.8] wherein the knowledgebase includes a neural network, wherein the one or more inputs are one or more input neurons of the neural network,

[85.9] wherein the machine readable code, when executed by the one or more processors, further causes the one or more processors to perform at least:

[85.10] generating a stream of circumstance representations comprising:

Substitute Claim 85

[85.11] the generated circumstance representation that (i) includes multiple object representation data structures that each comprises a label for object type, coordinates for object location, and a computer model for object size, and (ii) represents multiple objects detected at a first time, and

[85.12] another circumstance representation that (i) includes another multiple object representation data structures that each includes: a label for object type, coordinates for object location, and a computer model for object size, and (ii) represents the multiple objects detected at a second time,

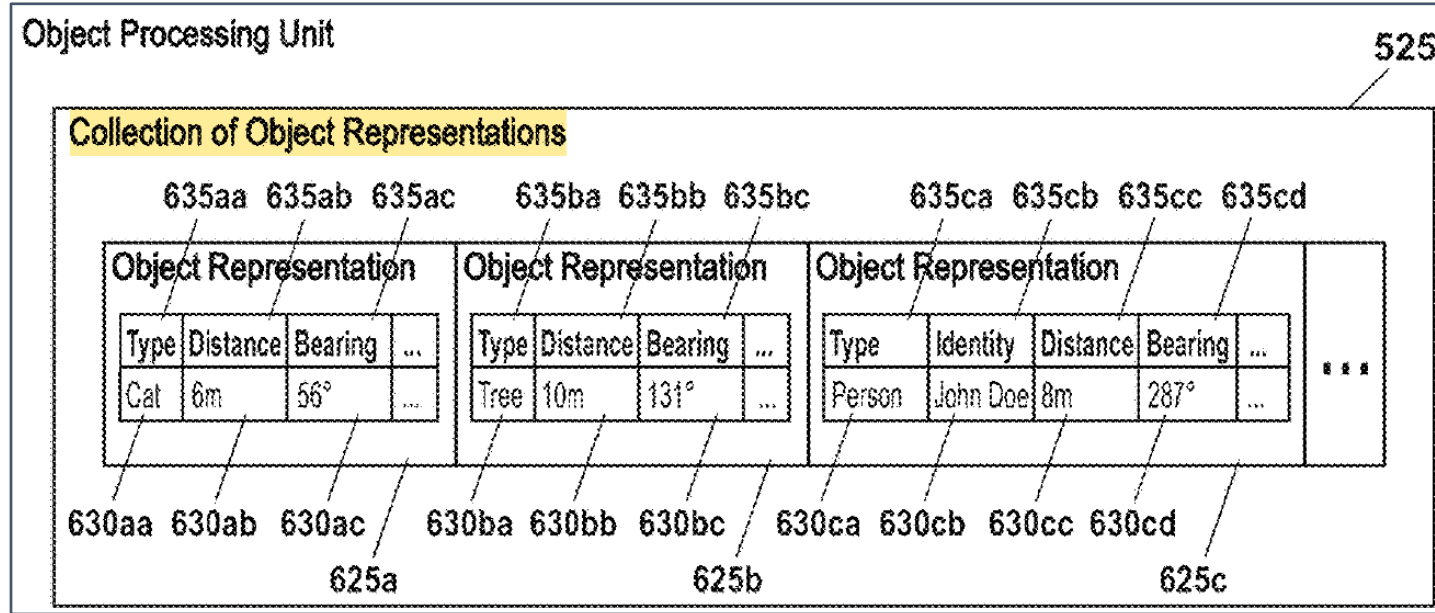
Substitute Claim 85

[85.13] wherein the determining the one or more instruction sets for operating the second device includes: inputting the multiple object representation data structures of the stream into the one or more input neurons, and using a correlation between the one or more input neurons and the one or more instruction sets for operating the second device, and

[85.14] determining another one or more instruction sets for operating a third device at least by: inputting the another multiple object representation data structures of the stream into the one or more input neurons, and using a correlation between the one or more input neurons and the another one or more instruction sets for operating the third device

- Proposed Substitute Claims
- **The Substitute Claims Have Adequate Written Description Support**
- Limitations [85.11]-[85.12]: Grotmol-Zhu Does Not Disclose or Suggest the Specifically Claimed Multiple Object Representation Data Structures
- Limitations [85.13]-[85.14]: Grotmol-Zhu-Buibas Does Not Disclose or Suggest Determining One or More Instruction Sets for a Second and Third Device by Inputting the Specifically Claimed Multiple Object Representation Data Structures into Input Neurons and Using a Correlation Between the Input Neurons and Instruction Sets

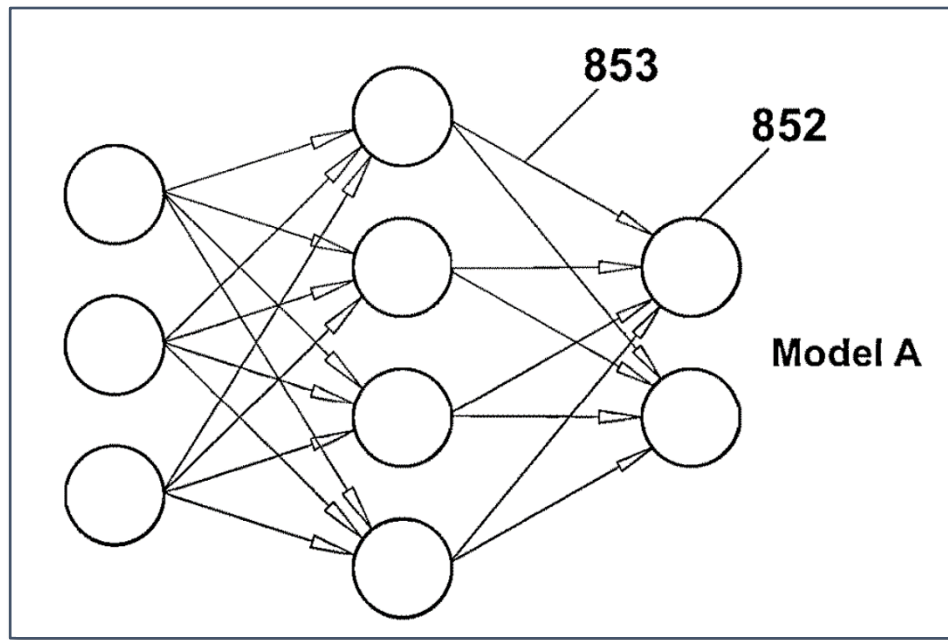
'474 Specification



FIGS. 4A-4B, illustrate an exemplary embodiment of Objects 615 detected in Device's 98 surrounding, and **resulting Collection of Object Representations 525**.

'976 Application

In general, object may include any object or part thereof that can be detected. Examples of object properties include existence of an object, **type of an object** (i.e. person, cat, vehicle, building, street, tree, rock, etc.), identity of an object (i.e. name, identifier, etc.), distance of an object, bearing/angle of an object, **location of an object** (i.e. distance and bearing/angle from a known point, **coordinates**, etc.), shape/**size of an object** (i.e. height, width, depth, **computer model**, point cloud, etc.), activity of an object (i.e. motion, gestures, etc.), and/or other properties of an object. Type of an object, for example, may include any classification of objects ranging from detailed such as



'474 Specification

15 programmed or learned function or method. For example, a neural network may include one or more input neurons that can be activated by inputs. Activations of these neurons can then be passed on, weighted, and transformed by a function to other neurons. Neural networks may range from those
20 with only one layer of single direction logic to multi-layer of multi-directional feedback loops. A neural network can use weights to change the parameters of the network's throughput. A neural network can learn by input from its environment or from self-teaching using written-in rules. A neural
25 network can be utilized as a predictive modeling approach in machine learning. An exemplary embodiment of a neural network (i.e. Neural Network 530a, etc.) is described later.

'991 Application

The disclosed artificially intelligent devices, systems, and methods for learning and/or using a device's circumstances for autonomous device operation comprise apparatuses, systems, methods, features, functionalities, and/or applications that enable learning a device's circumstances including objects with various properties along with correlated instruction sets for operating the device, storing this knowledge in a knowledgebase (i.e. neural network, graph, sequences, etc.), and operating a device autonomously. The disclosed artificially intelligent devices, systems, and methods for learning and/or using a device's circumstances for autonomous device operation, any of their elements, any of their embodiments, or a combination thereof can generally be referred to as DCADO, DCADO Unit, or as other suitable name or reference.

'991 Application

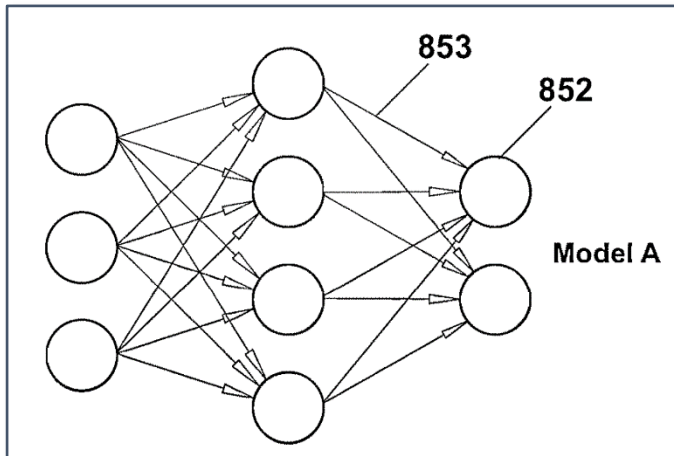
Referring to Fig. 12, an embodiment of Artificial Intelligence Unit 110 is illustrated. Artificial Intelligence Unit 110 comprises interconnected Knowledge Structuring Unit 520, Knowledgebase 530, Decision-making Unit 540, and Confirmation Unit 550. Other additional elements can be included as needed, or some of the disclosed ones can be excluded, or a combination thereof can be utilized in alternate embodiments.

as the intermediary data structures. In some embodiments, Knowledgebase 530 may be or include Neural Network 530a (later described). In other embodiments, Knowledgebase 530 may be or include Graph 530b (later described).

'991 Application

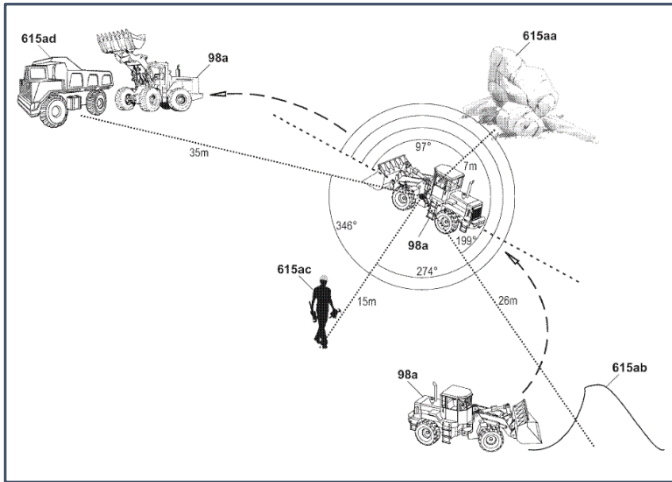
Further, Artificial Intelligence Unit 110 comprises the functionality for anticipating Device's 98 operation in circumstances including objects with various properties. Artificial Intelligence Unit 110 comprises the functionality for anticipating one or more instruction sets, data, and/or other information. Artificial Intelligence Unit 110 comprises the functionality for anticipating one or more Instruction Sets 526 based on one or more incoming Collections of Object Representations 525. Artificial Intelligence Unit 110 comprises the functionality for anticipating one or more Instruction Sets 526 to be used or executed in Device's 98 autonomous operation. Artificial Intelligence Unit 110 also comprises other disclosed functionalities.

'991 Application



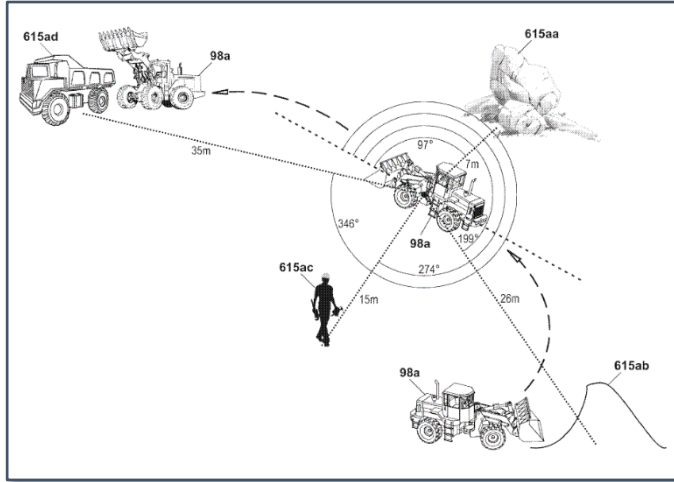
Further, Artificial Intelligence Unit 110 comprises the functionality for anticipating Device's 98 operation in circumstances including objects with various properties. Artificial Intelligence Unit 110 comprises the functionality for anticipating one or more instruction sets, data, and/or other information. Artificial Intelligence Unit 110 comprises the functionality for anticipating one or more Instruction Sets 526 based on one or more incoming Collections of Object Representations 525. Artificial Intelligence Unit 110 comprises the functionality for anticipating one or more Instruction Sets 526 to be used or executed in Device's 98 autonomous operation. Artificial Intelligence Unit 110 also comprises other disclosed functionalities.

'976 Application



Referring to Fig. 38, in some exemplary embodiments, Device 98 may be or include Loader 98a. Loader 98a may be operated by User 50 in person or remotely. Loader 98a may include or be coupled to one or more Sensors 92 (i.e. collectively referred to as Sensor 92, etc.) such as Camera 92a, Microphone 92b, Lidar 92c, Radar 92d, Sonar 92e, etc. and/or Object Processing Unit 93 that can detect Objects 615aa-615ad, and/or other elements or information in Loader's 98a surrounding. Object Processing Unit 93 may include Picture Recognizer 94a, Sound

Loader's 98a arm with bucket to collect a load, one or more motors or other actuators to move or maneuver Loader 98a, lifting system (i.e. hydraulic, pneumatic, mechanical, electrical, etc.) to lift a load, and/or arm with bucket to unload a load. Loader 98a may also include or be coupled to DCADO Unit 100. DCADO Unit 100 may be embedded (i.e. integrated, etc.) into or coupled to Loader's 98a Logic Circuit 250, Processor 11, and/or other processing element. DCADO Unit 100 may also be a program embedded (i.e. integrated, etc.) into or interfaced with Application



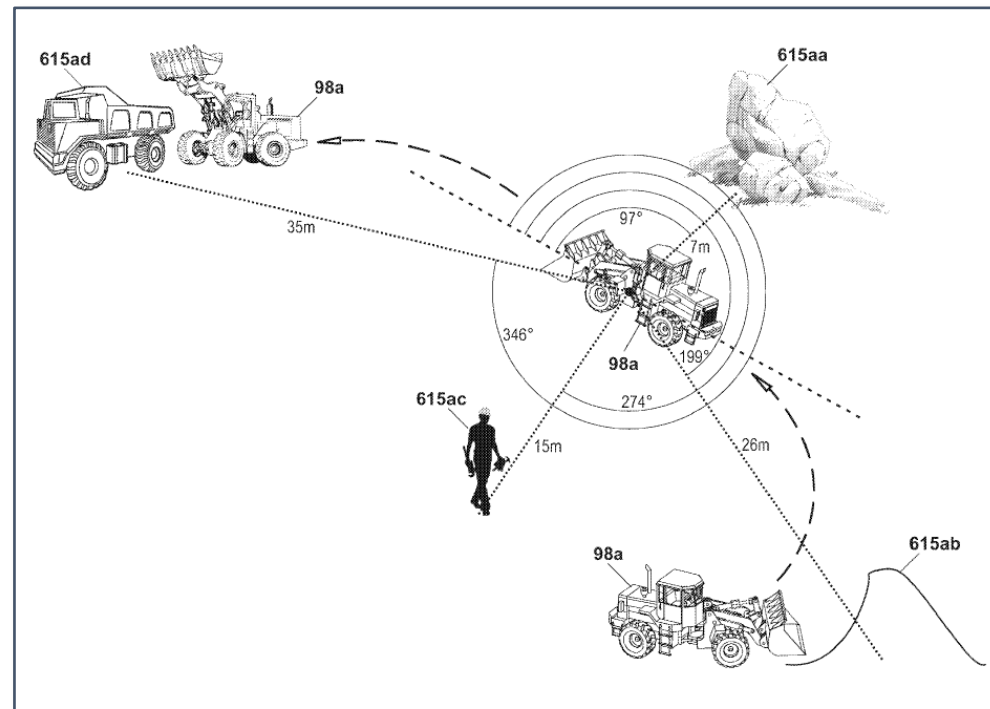
'976 Application

in previously learned ones. For instance, Loader 98a comprising DCADO Unit 100 may learn User 50-directed collecting, moving, maneuvering, lifting, unloading, and/or other operations in a circumstance that includes Rock 615aa, Pile of Material 615ab, Person 615ac, Truck 615ad, and/or other Objects 615 among which Loader 98a may need to maneuver and/or with which Loader 98a may need to interact. In the future, when a circumstance that includes Objects 615 with similar Object Properties 630 is encountered, Loader 98a may implement collecting, moving, maneuvering, lifting, and/or unloading operations autonomously.

In some embodiments, DCADO Unit 100 may reside on Server 96 accessible over Network 95 as previously described. In such embodiments, any number of Loaders 98a may connect to such remote DCADO Unit 100 and the remote DCADO Unit 100 may learn their operations in circumstances including objects with various properties. In turn, any number of Loaders 98a can utilize the remote DCADO Unit 100 for autonomous operation in circumstances including objects with various properties. For example, multiple operators (i.e. Users 50, etc.) may operate their Loaders 98a that are configured to transmit their operations in circumstances including objects with various properties to a remote DCADO Unit 100. Such remote DCADO Unit 100 enables learning of the operators' collective knowledge of operating Loaders 98 in circumstances including objects with various properties. Any number of Loaders 98 can utilize such collective knowledge comprised in the remote DCADO Unit 100 for their autonomous operation. Any of the disclosed elements such as Artificial Intelligence Unit 110, Knowledgebase 530, and/or others may reside on Server 96, and any combination of local and remote elements can be implemented in alternate embodiments.

configured to utilize the combined Knowledgebase 530. For example, a Device 98 including or interfaced with DCADO Unit 100 having access to a combined Knowledgebase 530 can use the collective knowledge learned from multiple Devices 98 for the Device's 98 autonomous operation.

'991 Application



- Proposed Substitute Claims
- The Substitute Claims Have Adequate Written Description Support
- **Limitations [85.11]-[85.12]: Grotmol-Zhu Does Not Disclose or Suggest the Specifically Claimed Multiple Object Representation Data Structures**
- Limitations [85.13]-[85.14]: Grotmol-Zhu-Buibas Does Not Disclose or Suggest Determining One or More Instruction Sets for a Second and Third Device by Inputting the Specifically Claimed Multiple Object Representation Data Structures into Input Neurons and Using a Correlation Between the Input Neurons and Instruction Sets

Substitute Claim 85

[85.11] the generated circumstance representation that (i) includes multiple object representation data structures that each comprises: a label for object type, coordinates for object location, and a computer model for object size, and (ii) represents multiple objects detected at a first time, and

[85.12] another circumstance representation that (i) includes another multiple object representation data structures that each includes: a label for object type, coordinates for object location, and a computer model for object size, and (ii) represents the multiple objects detected at a second time,

- “multiple object representation data structures”
 - Each comprising: “coordinates for object location, and a computer model for object size”

MTA Opposition

Ground	Substitute Claims	§ 103 Basis
MTA-1A	85-91	Grotmol-Zhu
MTA-1B	85-91	Grotmol-Zhu-Buibas

It would have been obvious to a POSITA to apply Zhu's suggestion for labeling the type of detected objects and defining bounding boxes for detected objects that provides a computer model for object size coordinates for object location to Grotmol for the reasons described above in the combination overview. TESLA-1008, 38:43-39:12; TESLA-1024, 11:16-36; *supra*, §II.B.1; Pet., 45-51, 73-77; TESLA-1022, ¶63.

Substitute Claim 85

[85.11] the generated circumstance representation that (i) includes multiple object representation data structures that each comprises: a label for object type, coordinates for object location, and a computer model for object size, and (ii) represents multiple objects detected at a first time, and

[85.12] another circumstance representation that (i) includes another multiple object representation data structures that each includes: a label for object type, coordinates for object location, and a computer model for object size, and (ii) represents the multiple objects detected at a second time,

and “model for “object size.” The object representation data structures are part of a circumstance representation, which represents a circumstance detected by one or more actual, real-world sensors (see Ex. 1001, Figs. 3A-3E, showing Camera 92a, Microphone 92b, Lidar 92c, Radar 92d, and Sonar 92e) for detecting objects and their properties, as recited in [85.2] and [85.5].

[85.2] one or more sensors;

* * *

[85.5] generating a circumstance representation, wherein the generated circumstance representation represents a circumstance detected at least in part by the one or more sensors;

Mot. to Amend [14] at Appx, 1

U.S. Patent

May 30, 2023

Sheet 3 of 40

US 11,663,474 B1

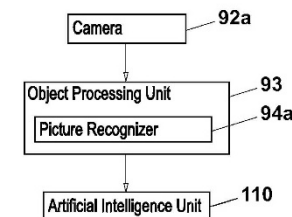


FIG. 3A

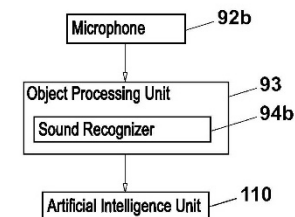


FIG. 3B

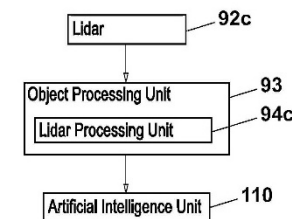


FIG. 3C

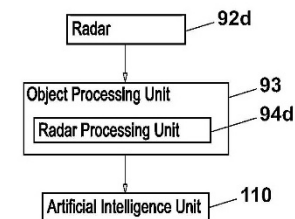


FIG. 3D

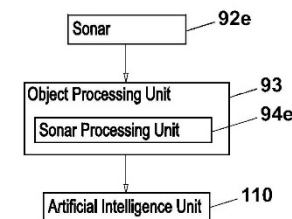
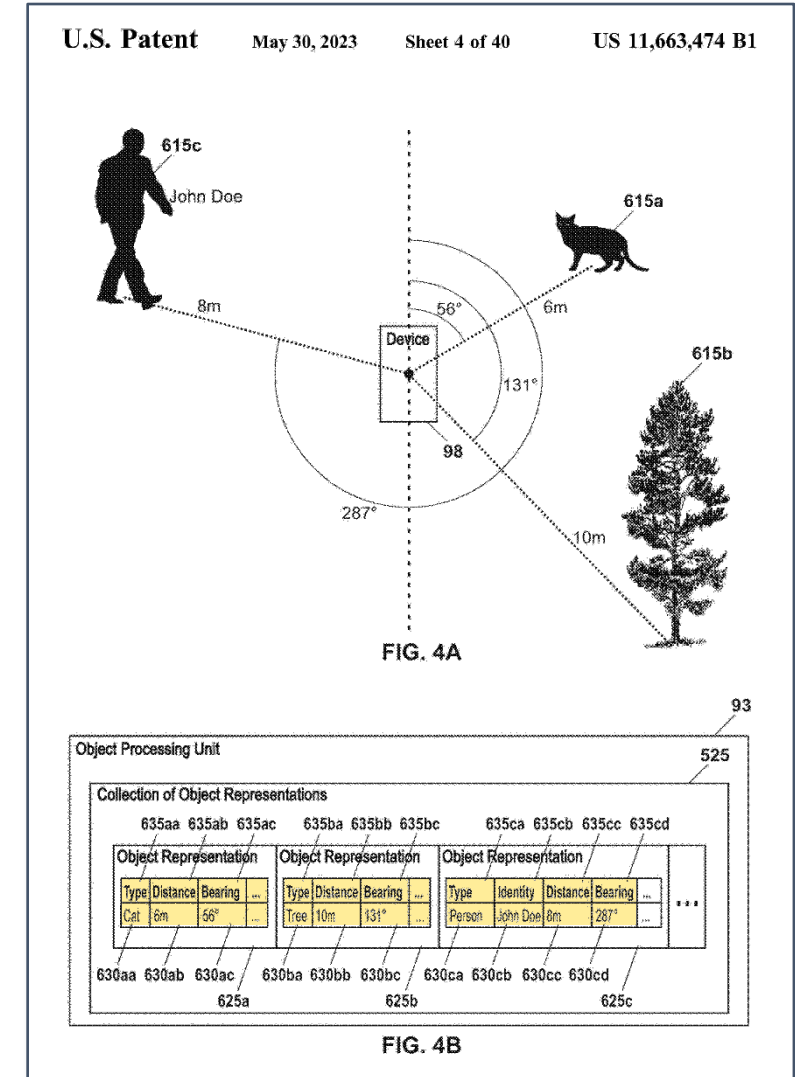


FIG. 3E

Dr. Saber Supplemental Declaration

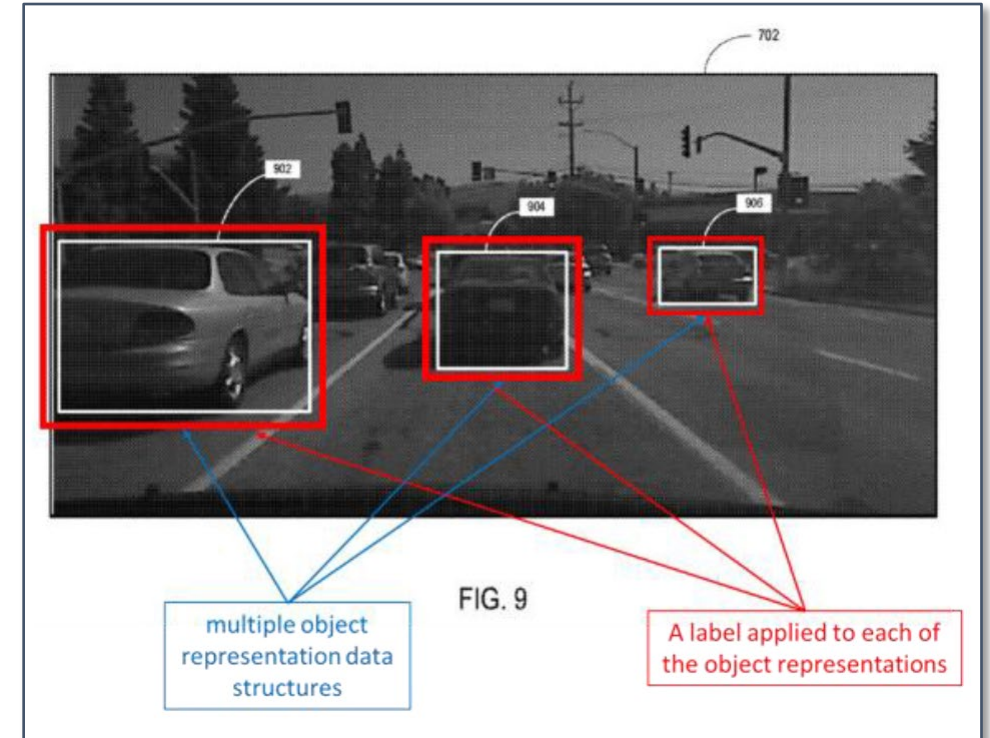
27. In my opinion, a POSA would have understood that the object (and its properties) that is detected by the sensor(s) is an **actual, real-world object** in the surroundings of the first device recited in [85.pre] that itself exists in the real world. Ex. 1001, 56:1-3 (“FIGS. 4A-4B, illustrate an exemplary embodiment of Objects 615 detected in Device's 98 surrounding, and resulting Collection of Object Representations 525.”), FIGS. 4A (showing an object 615a that is a cat at a distance of 6m from device 98), 4B (showing properties of the detected objects).



Dr. Saber Supplemental Declaration

28. In contrast, the Opposition relies on disclosures showing a bounding box in an image, the height and width of the box in pixels for the size of the bounding box (not the object's actual size), and first and second pixel coordinates that define location of the box in the image (not the object's actual coordinates). Opp., 11-12 (citing Ex. 1023, 14:43-67, 15:1-20, FIGS. 9-10). Petitioner annotated the bounding boxes as follows:

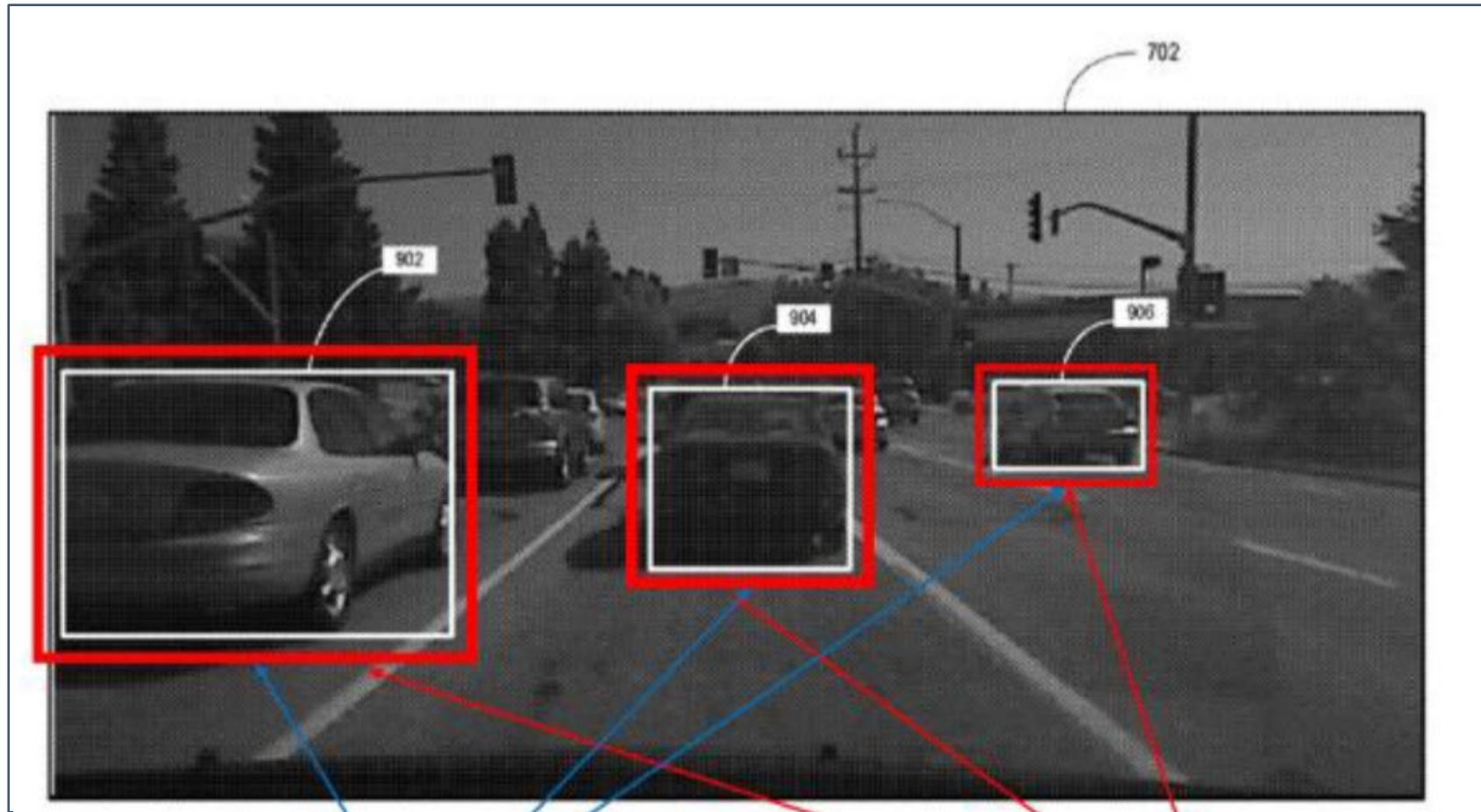
Ex. 2006 (Saber Supp. Decl.) at ¶28



Zhu

An object label may have one or more object label parameters that define the shape of the object label. The object label
55 parameters may be defined by one or more of the instructions
110, such as the vehicle detection algorithm 124. Moreover,
the object label parameters may vary depending on the sensor
type of the sensor that captured the image. For example,
where the shape of the object label is a bounding box, and the
60 sensor that captured the image is a one or more of the cameras,
the object label parameters may include a height parameter
that defines the height of the bounding box (e.g., in pixels), a
width parameter that defines the width of the bounding box
(e.g., in pixels), a first pixel coordinate that defines the lati-
65 tudinal placement of the bounding box (e.g., an X-coordi-
nate), and a second pixel coordinate that defines the longitu-
dinal placement of the bounding box (e.g., a Y-coordinate).

Zhu



- Proposed Substitute Claims
- The Substitute Claims Have Adequate Written Description Support
- Limitations [85.11]-[85.12]: Grotmol-Zhu Does Not Disclose or Suggest the Specifically Claimed Multiple Object Representation Data Structures
- **Limitations [85.13]-[85.14]: Grotmol-Zhu-Buibas Does Not Disclose or Suggest Determining One or More Instruction Sets for a Second and Third Device by Inputting the Specifically Claimed Multiple Object Representation Data Structures into Input Neurons and Using a Correlation Between the Input Neurons and Instruction Sets**

[85.13] wherein the determining the one or more instruction sets for operating the second device includes: inputting the multiple object representation data structures of the stream into the one or more input neurons, and using a correlation between the one or more input neurons and the one or more instruction sets for operating the second device; and

[85.14] determining another one or more instruction sets for operating a third device at least by: inputting the another multiple object representation data structures of the stream into the one or more input neurons, and using a correlation between the one or more input neurons and the another one or more instruction sets for operating the third device.

Mot. to Amend [14] at Appx, 2-3

- “determining ... instruction sets for operating the [second/third] device[s]”
 - “inputting the multiple object representation data structures ... into ... input neurons”

Grotmol

In some implementations of on-line learning for robot
40 navigation, the input vector x may be configured by concatenating the RGB values of the pixels in an image (e.g.,
obtained using video camera 166 in FIG. 1) and an additional 1-channel binary signal derived from the motor state.
The mandatory feature (e.g., the feature x_1 described above
45 with respect to FIG. 14) may be selected to comprise the 1-dimensional binary motor state.

Grotmol

As used herein the term **feature** may refer to a representation of an object edge, determined by change in color, luminance, brightness, transparency, texture, and/or curvature. The object features may comprise, inter alia, individual edges, intersections of edges (such as corners), orifices, and/or curvature

In some implementations, the **normalization** may comprise **shifting and/or scaling input features** to a given value range (e.g., A1=64 to A2=196 for an 8-bit pixel value, 0 to 1 range, and/or other range). In one or more implementations, the normalization may be configured based on determining an on-line estimate of the mean and standard deviation of feature values to obtain z-score for **individual feature (pixel)**. In one such implementation, for a given pixel (e.g., pixel at location (i1,i2)) a pair of values may be stored in history memory: one for the pixel mean and another for the pixel standard deviation. In some implementations, one or more parameters related to history of the input (e.g., pixel statistics) may be computed over a given interval, and/or the total duration of training. In one or more implementations, the learnings process may be configured to enable a user to reset contents of the parameter (e.g., pixel statistics).

In some implementations, data for one or more inputs may be scaled by a parameter NF, where NF is configured based on the overall number of features of a given feature types (i.e., the number of pixels in a subset t). In some implementations, the scaling parameter may be selected from the range between $\sqrt{NF}$ and $10 \times NF$.

In some implementations, **feature scaling operation** may comprise determining an average distance measure for a plurality of input feature instances (e.g., distance between 2-100 images for images acquired at 25 fps) and scaling the input in accordance with the average distance measure. Various scaling implementations may be employed, e.g., scaling the less redundant input, scaling the highly redundant input, and or combination thereof. The scaling operation may enable reducing disparity between contributions to the distance determination from a highly redundant input (e.g., video and/or other input) and less redundant input (e.g., audio, touch sensor, binary, and/or other input).

The feature scaling may be configured based on an observed and/or an expected characteristic or characteristics of a feature that may be salient to the action. By way of an illustration of an implementation of vision based robotic navigation, size of a target, e.g., number of pixels and/or cumulative pixel value corresponding to a ball 174 in FIG. 1, may be used to scale pixel values within a visual frame such that pixels of the target associated with the rescaled input may contribute comparably to the distance determination as a binary input feature (e.g., indicative of wheel rotation (or not), proximity sensor output (ON, OFF), battery level below threshold, and/or other binary input). In some implementations, the scaling configured based on observed and/or expected characteristic of a feature may be referred to as input equalization.

Dr. Saber Supplemental Declaration

etc.) as inputs to the neural network. *Id.*, 7:63-65, 30:54-31:35. Thus, image pixels or image features are used to provide inputs to Grotmol's neural network.

Grotmol does not disclose the claimed object representation data structures of [85.11]-[85.12] being inputted into input neurons. Opp., 14-15. For the claimed inputting of the object representation data structures into input neuron(s), the Opposition does not cite to Zhu or point to anything in Zhu, and does not even mention or cite Zhu for [85.13]-[85.14]. *Id.*, 14-16. Nevertheless, Zhu does not disclose or suggest inputting the claimed object representation data structures into input neurons. Thus, neither of the cited references (Grotmol and Zhu) actually discloses the claimed inputting of the specific claimed object representation data structures (recited in limitations [85.11]-[85.12]) into input neuron(s).

MTA Opposition

In the combination, it would have been obvious to a POSITA to input (i) the multiple object representation data structures for the objects detected at the first time (*supra*, [85.11]) into the input neurons of the neural network for the benefit of obtaining a first control output conditioned on the first input and (ii) the another object representation data structures for the objects detected at the second time (*supra*, [85.12]) into the input neurons of the neural network for the benefit of obtaining a second control output conditioned on the second input. TESLA-1008,



Second, in *Perfect Web*, the only case Appellees identifies in which common sense was invoked to supply a limitation that was admittedly *missing* from the prior art, the limitation in question was unusually simple and the technology particularly straightforward. “The patented

[7] Third, our cases repeatedly warn that references to “common sense”—whether to supply a motivation to combine or a missing limitation—cannot be used as a wholesale substitute for reasoned analysis and evidentiary support, especially when dealing with a limitation missing from the prior art references specified.

Arendi S.A.R.L. v. Apple Inc., 832 F.3d 1355, 1361 (Fed. Cir. 2016)

Dr. Saber Supplemental Declaration

34. In my opinion, it would have been complex, and certainly not unusually simple or straightforward, to replace one type of input (Grotmol's images or image features) into a neural network with another type of input (the object representation data structures of [85.11]-[85.12]) due to the complexities associated with a completely different structure of the neural network to accommodate completely different inputs, training with a different type of inputs, and the significant iterative nature of the training process that typically utilizes backpropagation type algorithms that are often combined with regularization techniques to appropriately train the network. This is so because, generally speaking during retraining, proper weights for correlations/connections are calculated/adjusted through the backpropagation process (*see* Ex. 1001, 108:64-67) for specific instruction sets associated with specific same inputs in each training set. Backpropagation, generally speaking, typically involves an iterative

Dr. Saber Supplemental Declaration

'474 Specification

back-propagation of errors can be implemented. Such back-propagations can then be used to adjust occurrence counts and/or weights of Connections **853** for better future predictions, for example. Any other back-propagation can be

Ex. 1001 ('474 pat.) at 108:64-67

training set. Backpropagation, generally speaking, typically involves an iterative process of training a neural network where a training input is provided to a given network and the corresponding calculated output is computed via feed forward operations through the network. The error between the calculated and expected output is then backpropagated throughout the network using algorithms such as gradient descent in order to properly update the weights throughout the layers. This process is repeated multiple times with various inputs using an appropriately chosen training dataset and proper regularization techniques (to ensure that the weights do not diverge and cause the overall network to become unstable) until convergence is achieved. The optimized network is then tested with hyperparameters (e.g., learning rates, line search, batch selection, etc.) potentially further optimized using a validation dataset and then further tested using a test dataset to measure/quantify its performance.

Dr. Saber's Supplemental Declaration

data structures of [85.11]-[85.12]. Petitioner's proposed modification of Grotmol's system would have resulted in an inoperable system, with incorrect

Grotmol's image-based neural network can only accept pixels of an image or potentially simple image features (such as normalization via feature shifting or scaling) to produce any meaningful results (e.g., a coordinate value or size data

input layer causing undue complexities in training given the enormous number of weights that would need to be handled and often yield erroneous, meaningless, and/or unstable results. Hence, neural networks are designed and trained for

of image pixels for instance). Introducing different style of inputs (e.g., object representations instead of pixels) into a neural network that is already trained for a specific task would potentially require a complete re-design of its layers, structure, redefinition of appropriate loss functions to quantify the differences between desired and actual outcomes, selection of a properly diversified training dataset and many more followed by a complete retraining with appropriate use of

36. It would not have been predictable to just replace one type of input with a different type of input as Petitioner and Dr. Janet assert. Rather, it was