

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

REALTEK SEMICONDUCTOR CORP.,
Petitioner,

v.

ATI TECHNOLOGIES ULC
Patent Owner.

Case No. IPR2023-00564
U.S. Patent No. 7,742,053

**DECLARATION OF JOHN HART REGARDING
U.S. PATENT NO. 7,742,053**

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. QUALIFICATIONS AND EXPERIENCE.....	2
III. MATERIALS CONSIDERED	4
IV. LEGAL STANDARDS	6
A. Claim Construction.....	7
B. Obviousness	7
V. CLAIM CONSTRUCTION	9
VI. RELEVANT TECHNICAL BACKGROUND AT THE TIME OF THE '053 PATENT	12
VII. U.S. PATENT NO. 7,742,053	14
VIII. DESCRIPTION OF ASSERTED ART	17
A. Stuttard (PCT Pub. No. WO00/62182A2).....	17
1. Overview.....	17
2. Phased Architecture	22
B. Williams (PCT Pub. No. WO00/63770A1).....	24
C. Whittaker (U.S. Patent No. 5,968,167)	25
IX. EVALUATION OF PETITIONER’S PROPOSED GROUNDS	25
A. Ground 1: Stuttard and the Knowledge of a POSITA Does Not Render Claims 1-9 of the '053 Patent Obvious	25
1. Stuttard Does Not Disclose or Render Obvious the Claimed “at least one memory device”.....	26
2. Stuttard Does Not Disclose or Render Obvious Storing Command Threads	31
3. Stuttard’s Alleged Arbiter Does Not Select Between a Plurality of Pixel and Vertex Command Threads	32
B. Ground 2: Stuttard, Williams, and the Knowledge of a POSITA Does Not Render Claims 1-9 of the '053 Patent Obvious.....	35

1.	Williams Does Not Disclose Or Render Obvious A Memory Device That Stores A Plurality Of Pixel Command Threads And Vertex Command Threads At The Same Time.....	35
2.	There Is No Motivation to Combine Williams with Stuttard.....	43
C.	Ground 3: Williams, Whittaker, and the Knowledge of a POSITA Does Not Render Claims 1-9 of the '053 Patent Obvious.....	46
1.	Williams Does Not Disclose or Render Obvious Storing Command Threads	46
2.	Williams Does Not Disclose Or Render Obvious An Arbiter That Selects A Command Thread	46
3.	Williams Does Not Disclose Or Render Obvious Command Processing Engines	47
4.	Neither Williams Nor Whittaker Discloses Or Render Obvious An Arbiter That Selects Command Threads Based On Relative Priorities Of The Command Threads.....	49
5.	There is No Motivation to Combine Williams with Whittaker, Nor Any Reasonable Expectation of Success.....	52
X.	CONCLUSION.....	53

I, John Hart, being of legal age, hereby declare, affirm, and state the following:

I. INTRODUCTION

1. I have been retained by ATI Technologies UCL (“ATI” or “Patent Owner”), as an independent expert in this proceeding before the Patent Trial and Appeal Board (“PTAB” or “Board”). I understand that Realtek is requesting that the Board institute an Inter Partes Review (“IPR”) proceeding of U.S. Patent No. 7,742,053 (the “’053 Patent”) (Ex. 1001) in IPR2023-00564.

2. I am providing this testimony for consideration by the Patent Trial and Appeal Board in this proceeding. Many of my opinions are also responsive to the Declaration of Massoud Pedram (“Pedram Decl.”) (Ex. 1004), which I understand was submitted in support of Realtek’s Petition for *Inter Partes* Review of the ’053 Patent. I have personal knowledge of the facts and opinions set forth in this declaration and all of the opinions and conclusions found in this declaration are my own.

3. I am being compensated at my standard hourly rate for my services. I have no personal or financial stake or interest in the outcome of the present proceedings, and my compensation is not dependent in any way upon the outcome of this proceeding.

II. QUALIFICATIONS AND EXPERIENCE

4. I have been a professor of computer science, focusing on computer graphics, since 1992. I am currently a tenured Professor in the Department of Computer Science at the University of Illinois at Urbana-Champaign, a department consistently ranked in the top-5 by US News and World Report. I am also Director of the Office for Professional Education of the University of Illinois at Urbana-Champaign.

5. I received a Bachelor's of Science degree in Computer Science from Aurora University in 1987, a Master's of Science in Electrical Engineering and Computer Science from the University of Illinois at Chicago in 1989, and a Ph.D. in Electrical Engineering and Computer Science from the University of Illinois at Chicago in 1991.

6. I worked as an intern in 1998 at the IBM TJ Watson Research Center in Hawthorne, New York, and in 1999 at AT&T Pixel Machines in Holmdel, New Jersey. I worked as a postdoc in 1991-1992 at the Electronic Visualization Laboratory at the University of Illinois at Chicago with funding from the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign. I worked as a visiting researcher in the summer of 2007 for the Graphics Research Group of Adobe Systems in Seattle, Washington.

7. From 1997-2000, I designed, simulated and patented graphics hardware for Silicon Reality and the Evans & Sutherland Computer Corp. I have also consulted for visual effects companies (e.g., Kleiser-Walczak, Blue Sky | VIFX), defense contractors (e.g., SAIC, Pratt & Whitney) and medical imaging companies (e.g., Intrinsic Medical Imaging).

8. My research in computer graphics has been supported by Adobe, DARPA, Intel, Nokia, NVIDIA, NSF and Microsoft. I have published over 75 reviewed papers on computer graphics, including papers in the most rigorous and prestigious venues, including the SIGGRAPH Annual Conference Proceedings, the ACM Transactions on Graphics, and the IEEE Transactions on Visualization and Computer Graphics. From 2002-2008, I was the longest-serving Editor-in-Chief of ACM Transactions on Graphics, the top journal in the field of computer graphics.

9. I was an Executive Producer for the ACM SIGGRAPH documentary “The Story of Computer Graphics” in 1999. I served from 1996-2000 on the ACM SIGGRAPH Executive Committee. I served as the founding computer graphics area editor of the ACM Books series, and oversaw the publication of “The VR Book: Human Centered Design for Virtual Reality.”

10. I co-authored the book “Real-Time Shading” in 2002 on the implementation of procedural shaders on graphics processors became fully user programmable GPU’s. I contributed two chapters to the book “Modeling and

Texturing: A Procedural Approach” in 2002, including a chapter on procedural texturing using rasterization and texturing operations.

11. I am being compensated for my time working on this matter at my usual rate of \$575 per hour. My compensation does not depend on the outcome of this litigation.

12. I believe I meet the criteria for a POSITA as laid out in Investigation No. 337-TA-1318, Ex. 1010 at 9, and any other appropriate definition of a POSITA for this patent.

13. Additional details about my employment history, fields of expertise, and publications are further described in my curriculum vitae, which is attached as Exhibit A to this report.

III. MATERIALS CONSIDERED

14. In forming the opinions set forth herein, I have considered and relied upon my education, knowledge in the relevant field, and my experience. I have also reviewed and considered the materials cited herein, including the following materials:

Exhibit No. IPR2023-00564	
Ex. 1001	U.S. Patent 7,742,053 to Lefebvre et al.
Ex. 1002	Prosecution History of U.S. Patent 7,742,053
Ex. 1004	Declaration of Massoud Pedram, Ph.D.
Ex. 1005	PCT Pub. No. WO00/62182A2 to Stuttard et al.
Ex. 1006	PCT Pub. No. WO00/63770A1 to Williams et al.

Exhibit No. IPR2023-00564	
Ex. 1007	U.S. Patent 5,968,167 to Whittaker et al.
Ex. 1008	Curriculum Vitae of Massoud Pedram, Ph.D.
Ex. 1010	<i>Certain Graphics Systems, Components Thereof, and Digital Televisions Containing The Same</i> , Inv. No. 337-TA-1318 (Order No. 31, Oct. 18, 2022)
Ex. 1011	<i>LG Elecs., Inc. v. ATI Techs. ULC</i> , IPR2015-00325, Paper 62 (Final Written Decision) (PTAB Apr. 14, 2016)
Ex. 1012	U.S. Patent 7,116,332 to Boyd et al.
Ex. 1013	U.S. Patent 7,038,685 to Lindholm et al.
Ex. 1014	H. Selzer, Dynamic Load Balancing Within a High Performance Graphics System, Proceedings of the Sixth Eurographics Conference on Advances in Computer Graphics Hardware: Rendering, Visualization and Rasterization Hardware (Sept. 1, 1991)
Ex. 2002	Certain Graphics Systems, Components Thereof, and Digital Televisions Containing The Same, Inv. No. 337-TA-1318, Final Initial Determination (July 7, 2023)
Ex. 2003	James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. 1990. Computer Graphics: Principles And Practice (2nd ed.). Addison-Wesley Longman Publishing Co., Inc
Ex. 2004	Akenine-Möller, T., Haines, E., & Hoffman, N. (2008). Real-Time Rendering (3rd ed.). A K Peters/CRC Press.
Ex. 2005	Certain Graphics Systems, Components Thereof, and Digital Televisions Containing The Same, Inv. No. 337-TA-1318, Hearing Transcript, Day 4 (March 16, 2023)
Ex. 2006	Edwin Earl Catmull. 1974. "A Subdivision Algorithm For Computer Display of Curved Surfaces." Ph.D. Dissertation. The University of Utah
Ex. 2007	Difference Between Buffering https://techdifferences.com/difference-between-buffering-and-caching-in-os.html

Exhibit No. IPR2023-00564	
Ex. 2008	Certain Graphics Systems, Components Thereof, and Digital Televisions Containing The Same, Inv. No. 337-TA-1318, Deposition of Dr. Pedram (January 04, 2023)

15. This declaration is made based on information currently available to me. I intend to continue my investigation and study, which may include a review of documents and information that may yet be produced, as well as deposition testimony from depositions for which transcripts are not yet available or that may yet be taken in this proceeding. Therefore, I expressly reserve the right to expand or modify my opinions as my investigation and study continue, and to supplement my opinions in response to any additional information that becomes available to me, whether from Petitioner (or its experts), Patent Owner (or its experts), or in light of any relevant opinions or orders from the Patent Trial and Appeal Board or other authoritative body.

IV. LEGAL STANDARDS

16. In preparing and expressing my opinions and considering the subject matter of the '053 Patent, I am relying on certain basic legal principles that counsel have explained to me. These principles are discussed below.

A. Claim Construction

17. I understand that the first step in determining the validity of a claim is for the claim to be properly construed. I have been further advised that, in an *inter partes* review proceeding, the claims of a patent are typically given their ordinary and customary meaning as would be understood by one of ordinary skill in the art in the context of the claims and the patent disclosure (specification).

18. I further understand that an inventor can provide special definitions for specific claim term(s) within the patent. However, any special definition for a claim term must be set forth in the specification with reasonable clarity and precision.

B. Obviousness

19. I have been informed that a claim may be invalid as obvious if the subject matter described by the claim as a whole would have been obvious to a POSITA at the time the claimed invention was made. I understand that the standard for obviousness in an *inter partes* review proceeding is by a preponderance of the evidence.

20. I have also been informed that a determination of obviousness involves an analysis of the scope and content of the prior art, the similarities between the claimed invention and the prior art, and the level of ordinary skill in the art. I have been informed and understand that a prior art reference should be viewed as a whole.

21. I have been informed that in considering whether an invention for a claimed combination would have been obvious, I may assess whether there are apparent reasons to combine known elements in the prior art in the manner claimed in view of interrelated teachings of multiple prior art references, the effects of demands known to the design community or present in the market place, and/or the background knowledge possessed by a POSITA. I also understand that other principles may be relied on in evaluating whether a claimed invention would have been obvious.

22. I have been informed that, in making a determination as to whether or not the claimed invention would have been obvious to a POSITA, one may consider certain objective indicators of non-obviousness if they are present, such as: commercial success of product(s) practicing the claimed invention; long-felt but unsolved need; teaching away; unexpected results; copying; and praise by others in the field. I understand that for such objective evidence to be relevant to the non-obviousness of a claim, however, there must be a causal relationship (called a “nexus”) between the claim and the evidence. I also understand that this nexus must be based on a novel element of the claim rather than something available in the prior art.

23. I have also been informed and understand that when considering the obviousness of a patent claim, one should consider whether a reason or motivation

existed for combining the elements of the references in the manner claimed, and that the prior art must create a reasonable expectation of success in producing the claimed subject matter.

V. CLAIM CONSTRUCTION

24. I understand certain terms of the '053 Patent have been previously construed by the Board and by the ALJ in the ITC Investigation. The ALJ construed certain terms in a claim construction order (Ex. 1010) during the pendency of the investigation. He further construed other terms still in dispute after the evidentiary hearing in his initial determination. Ex. 2002 (ID) at 32-43.

25. The table below summarizes the Board's adopted construction of certain terms.

Claim Term	The Board's Adopted Construction
"arbiter"	"any computer hardware, software, or combination thereof that receives and provides acommand thread" Ex. 1011 at 7.
"command processing engine"	"as requiring each command processing engine be coupled to an arbiter and operable to receive and process a command thread selected by the arbiter" Ex. 1011 at 12.

26. The table below summarizes the ALJ's adopted construction of certain terms in his claim construction order.

Claim Term	ALJ's Adopted Construction in Claim Construction Order
"arbiter," "select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads," and "relative priorities"	"Plain and ordinary meaning." Ex. 1010 at 18-20.
"command threads"	"'sequence of commands applicable to the corresponding element,' with no requirement that they comprise state bits." Ex. 1010 at 22-23.
"memory device"	"Respondents' proposed construction of 'memory device' is rejected, which resolves the parties' dispute, and the term is otherwise not construed." Ex. 1010 at 23-24.
"coupled to"	"Respondents' proposed construction of 'coupled to' is rejected, which resolves the parties' dispute, and the term is otherwise not construed." Ex. 1010 at 24-25.

27. Out of the above constructions, I understand Dr. Pedram has applied the narrower of the available constructions for terms that have more than one construction. Ex. 1004, ¶ 42. I have done the same in my analysis.

28. The table below summarizes the ALJ's adopted construction of certain terms in his initial determination, which was released after this petition was filed. I have applied the following constructions in my analysis.

Claim Term	ALJ's Adopted Construction in Initial Determination
"at least one memory device...operative to store a plurality of pixel command	"[A]t least one memory device...operative to store a plurality of pixel command threads and...a plurality of vertex command threads"

Claim Term	ALJ's Adopted Construction in Initial Determination
threads and...a plurality of vertex command threads"	means operative to store a plurality of complete sequences of pixel commands and a plurality of complete sequences of vertex commands. Ex. 2002 (ID) at 39. "[I]t is a correct claim construction that 'at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads' means the memory device(s) are operable to store both pluralities at the same time." Ex. 2002 (ID) at 84.
"select a command thread"	"[S]elect a command thread does not mean selecting one and only one thread. Ex. 2002 (ID) at 41.
"command processing engine"	"[A] person of ordinary skill would have understood a "command processing engine" could be implemented in software, hardware, or a mix of the two." Ex. 2002 (ID) at 43.

29. Additionally, I agree that the claimed "memory device" must be operative to store pluralities of pixel command threads and vertex command threads at the same time. As rationale, both claim 1 and claim 5 require that the arbiter "select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads." Ex. 1001 at Claims 1, 5. In my opinion, the arbiter can only perform the claimed selection when both pluralities of command threads are stored at the same time. Therefore, the plain and ordinary meaning of "at least one memory device . . . operative to store a plurality of pixel command

threads and . . . a plurality of vertex command threads” requires storing complete command threads as well as storing a plurality of pixel command threads and a plurality of vertex command threads at the same time.

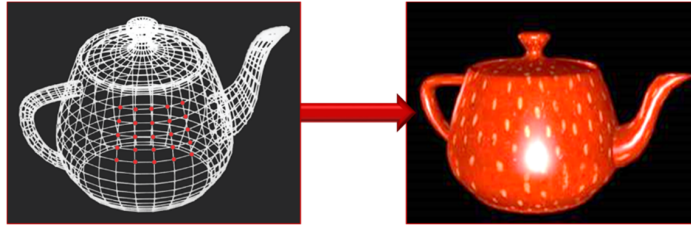
30. For all other claim terms, I applied their plain and ordinary meaning as it would be understood by a POSITA.

VI. RELEVANT TECHNICAL BACKGROUND AT THE TIME OF THE '053 PATENT

31. To the extent it may be helpful to the Board in understanding the context of the '053 Patent and in framing my opinions, I offer the following summary of pertinent technical information that would have been available to a person of ordinary skill in the art as of August 24, 2001, which I understand is the earliest priority date for the '053 Patent and is thus the lens through which the subject matter is to be evaluated.

32. Graphics processing involves manipulating and rendering 3D graphical objects onto a 2D display screen. Ex. 2004 (Real-Time Rendering) at 11. Specifically, 3D objects (typically represented by a collection of “primitives,” geometrical shapes such as triangles) are converted into a 2D image (typically comprising a collection of pixels) as shown below. Ex. 2003 (Foley) at 3, 11-15, 17-20; Ex. 2004 at 11-13. The 2D image represents a depiction of the 3D objects, rendered from the perspective of a given viewer, and shaded to convey the object's

surface material illuminated from a given light source. Ex. 2003 at 210-222, 722-741; Ex. 2004 at 21-27.



33. The conversion of a 3D shape consisting of a mesh of triangles into a photorealistic 2D image consisting of an array of pixels happens in a pipeline of stages. Ex. 2003 at 806-809, 866-871; Ex. 2004 at 12. The first stage is often referred to as “vertex processing” and computes, for the 3D position in space of each vertex (the point in space corresponding to the junction where neighboring triangles meet), a new 2D position of that vertex in the space of the screen or output image from the perspective of the viewer. Ex. 2003 at 806-809, 866-871; Ex. 2004 (Real-Time Rendering) at 14-21.

34. The next stage, “scan conversion,” begins the “rasterization” process by converting the 2D triangle described by the three 2D vertices output from the vertex processing stage, into the pixels on the screen that lie within that triangle. Ex. 2003 at 14, 806-809. This the scan conversion stage emits all the screen-space pixels that are covered by the triangle. Ex. 2003 at 806-809; Ex. 2004 at 21-22. The final “pixel processing” stage computes the colors of each of these pixels/fragments, and

applies textures. Ex. 2003 at 868-872; *see generally* Ex. 2006 (Catmull); Ex. 2004 at 22-23.

VII. U.S. PATENT NO. 7,742,053

35. The '053 Patent is directed to graphics processing systems, and specifically towards the processing of multiple command threads in parallel. The '053 Patent disclosed novel methods for processing multiple threads across one or more hardware processing units with the specialized parallel processing architecture of a graphics processor.

36. For example, in a graphics processing system, the management and control of multiple command threads relating to texture applications is important. Ex. 1001 at 1:22-24. Typically, multiple steps are required to process elements for the application of textures and other processing structures. *Id.* at 1:24-28. A preferred manner for improving the efficiency of graphics processing systems is controlling the flow of the multiple command threads. *Id.* at 1:28-31.

37. Prior art systems were rigid systems that required providing command threads sequentially. *Id.* at 1:31-48. Another consequence of this architecture was the lack of support for an unlimited number of dependent texture fetches. *Id.* at 2:2-6. Prior to the '053 Patent, conventional systems consisted of two separate command processing engines: a pixel command processing engine that only ran pixel commands and a vertex command processing engine that only ran vertex commands.

The vertex command processing engine would first process vertices. The triangle primitives defined by these processed vertices would then be scan converted into the pixels that filled them in. These pixels would then be processed by the pixel processing engine to determine their final color. Since there were 10x to 100x more pixels than vertices, graphics processing could run into a load imbalance causing one of the command processing engines to wait while the other finished its processing.

38. The '053 Patent solves those problems with a novel architecture that uses a single “unified shader processor” (the “command processing engine”) capable of executing both pixel commands and vertex commands, along with the necessary arbitration to determine how to efficiently process commands given that there were more pixel processing commands than vertex processing commands:

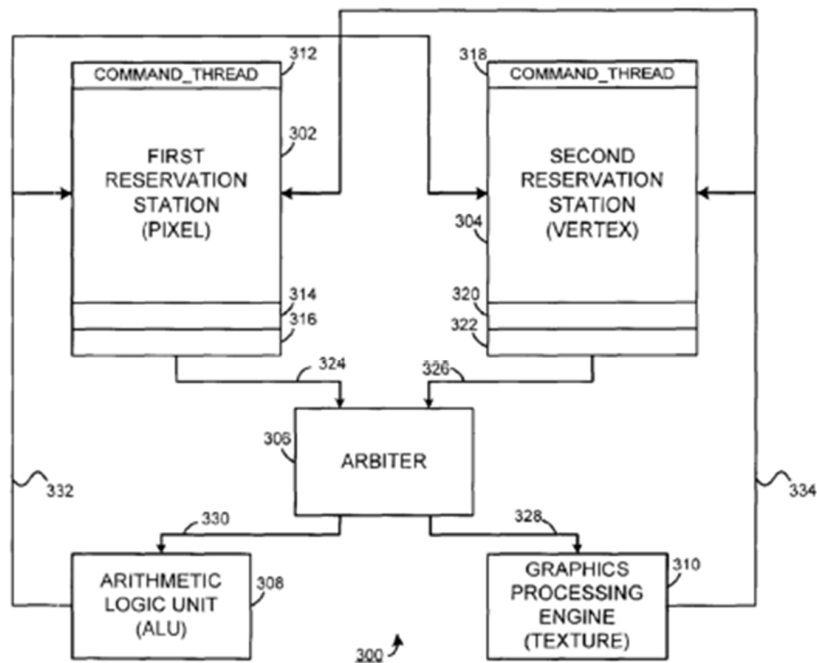


FIG. 4

Id. at Fig. 4. As a result, the arbiter 309 in Fig. 4 above could ensure that the unified shader processor was kept as busy as possible while it completed rendering of the current frame as quickly as possible, *e.g.*, either processing the final pixels of the current frame, or by processing the vertices for triangles in the next frame. *Id.* at 3:16-20.

39. The “command processing engine” (*i.e.*, the “unified shader processor”) consists of specialized resources, such as an ALU and a texture unit. *Id.* at 2:59-62. The ’053 Patent specification teaches that these resources include parallel processing. *Id.* at 3:40-44. For example, Fig. 3 shown below illustrates an example where the command processing engine includes multiple ALU’s 232 and thread

sequencer logic 236 designed to further optimize the parallel execution of command threads. *See, e.g., id.* at 3:37-47.

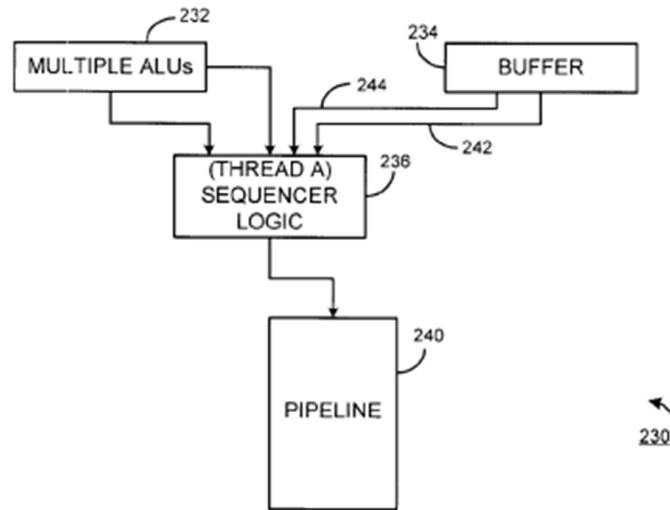


FIG. 3

Id. at Fig. 3.

VIII. DESCRIPTION OF ASSERTED ART

A. Stuttard (PCT Pub. No. WO00/62182A2)

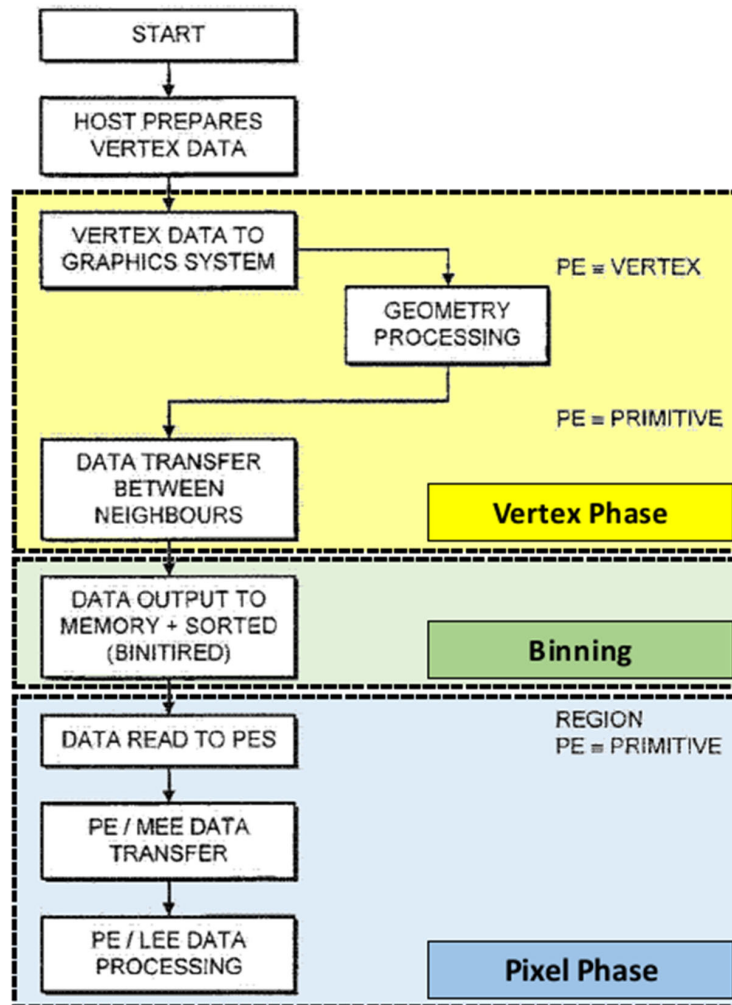
1. Overview

40. Stuttard (Ex. 1005) is directed to a system which attempts to improve the use of a “single-instruction, multiple data” (“SIMD”) architecture by dividing the processing elements into processing blocks. Ex. 1005 at 2:1-24. Specifically, Stuttard discloses “*a* SIMD . . . array of processing elements, wherein the processing elements are operably divided into a plurality of processing blocks, the processing

blocks being operable to process respective groups of data items.” *Id.* at 2:19-23 (emphasis added).

41. In Stuttard, however, all of the processing blocks of the SIMD operate on “***a common received instruction.***” *Id.* at 2:25-34; *see also id.* at 11:33-35 (“[T]he ***same instruction stream*** is supplied to all of the processing blocks in the system.”). Thus, at any given time, all processing units within Stuttard are operating on the same instruction. This is distinct from the parallel approach in the ’053 patent, described above.

42. Also critical to Stuttard’s architecture is that it operates first exclusively on vertices in one phase, and only when that phase is fully complete, then operates exclusively on pixels in a later phase, with no overlap in the execution of these two phases. As explained more fully below, when Stuttard is operating in one phase, it does not store the alleged command threads relating to the other phase in memory (in particular in the small, fast cache memory Dr. Pedram focuses on) and would have no reason to be stored there at that time.



Id. at Fig. 9 (annotated).

43. The architecture of Stuttard is illustrated and described in Fig. 1 reproduced below. As shown in Figure 1, Stuttard discloses that the system interfaces with a host system (not shown) via the interface 2. The graphics system 3 includes an embedded processing unit (EPU) 8 “for controlling the overall function of the graphics processor and for interfacing with the host system.” *Id.* at 4:26-30.

The graphics processing capability of Stuttard's system is provided by a processing core 10. *Id.* at 7:25-27.

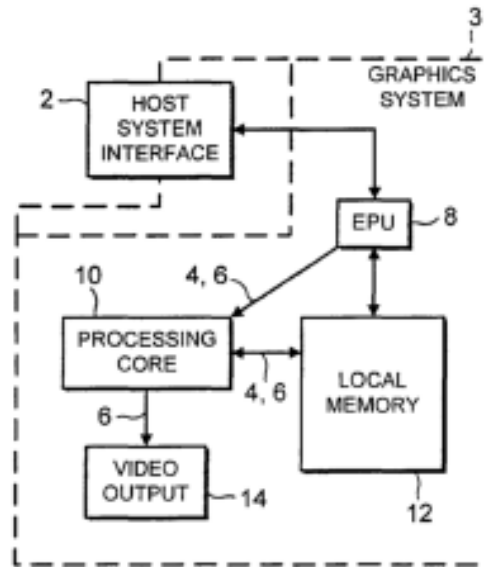


FIG. 1

Id. at Fig. 1.

44. The processing core includes processing blocks 106, which comprise an array 1061 of processor elements (PEs), a mathematical expression evaluator (MEE), and a binning unit. *Id.* at 8:15-9:3. To interface with and manage thread execution by the processing core, Stuttard describes the use of a thread manager 102, an array controller 104, and a channel controller 108, which are illustrated and describe in connection with, for example, Figure 3 (reproduced below).

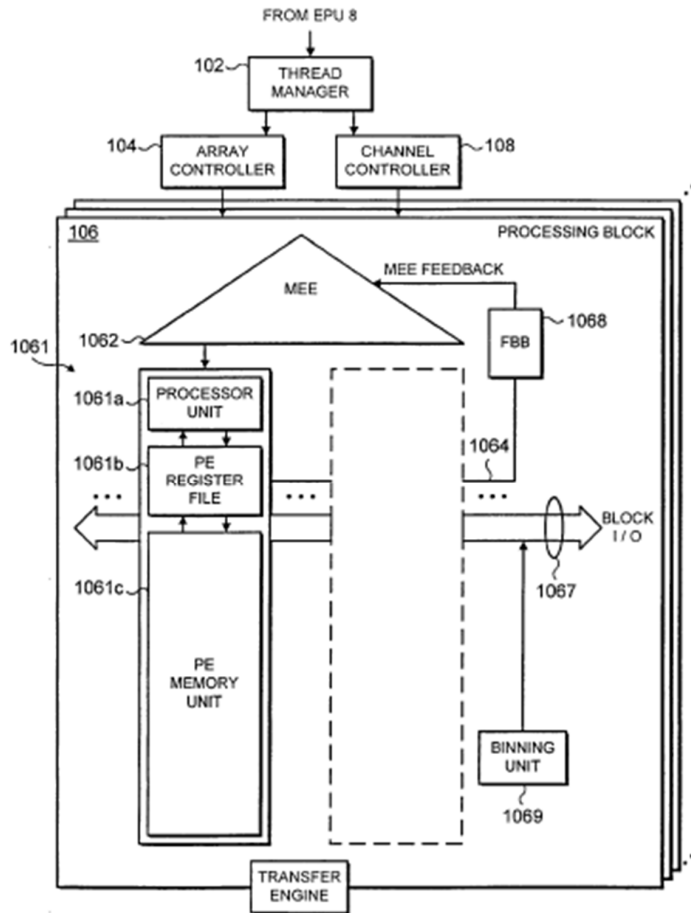


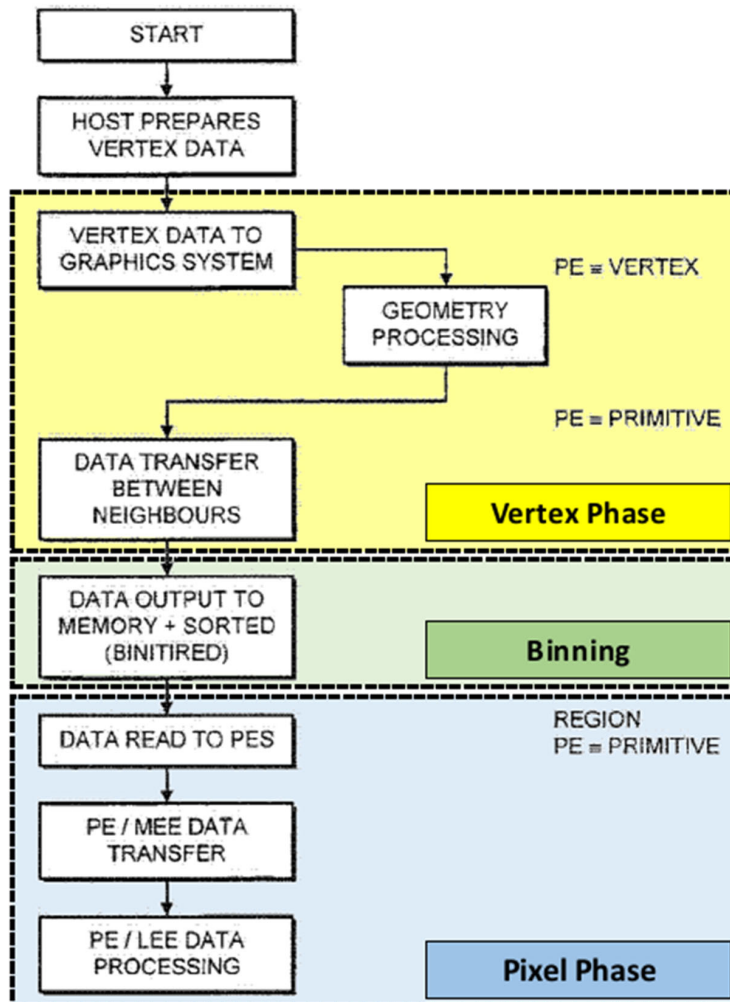
FIG. 3

Id. at Fig. 3. “The thread manager 102 is connected to receive control signals from the EPU 8. The control signals inform the thread manager as to when instructions are to be fetched and where the instructions are to be found. The thread manager 102 is connected to provide these instructions to the array controller 104 and to the channel controller 108. The array and channel controllers 104 and 108 are connected to transfer control signals to the processing blocks 106 dependent upon the received instructions.” *Id.* at 8:17-27. Stuttard further discloses that “[e]ach processing block

includes a binning unit 1069” and that “[t]he purpose of the binning unit 1069 is to sort primitive data by region, since the data is generally not provided by the host system in the correct order for region based processing.” *Id.* at 8:34-35; 14:30-35.

2. Phased Architecture

45. Stuttard is a phased system in which all vertex data and command threads are stored and processed before any pixel data and command threads can be stored and processed:



Id. at Fig. 9 (annotated).

46. Figure 9 of Stuttard provides steps for processing graphics data using the system. *Id.* at 20:23-25, Fig. 9. The host system prepares data concerning the vertices of the primitive graphical images to be processed and displayed by the system and transfers the data either as a block of vertex data, or vertex by vertex. *Id.* at 20:25-30. The data is loaded into the PEs of the graphics system so that each PE contains data for one vertex, and each PE represents a vertex of a primitive. *Id.* at 20:32-21:1. The received data is processed to transform it from the host system reference space to the required screen space. *Id.* at 21:3-13.

47. The primitive data is then output from the PEs to local memory so that it can be sorted by region by the binning unit. *Id.* at 21:15-21. The binning unit provides a hardware implemented region sorting system which removes the sorting process from the processing elements, thereby releasing the PEs for data processing. *Id.* at 21:23-26. After the primitive data is sorted, it is read back into the individual PEs, so all the PEs in one processing block contain data concerning respective primitives occurring in a single region. *Id.* at 22:4-10. Next, the PE transfers its data concerning its primitive to the MEE for scan conversion. *Id.* at 12-24. Only after scan conversion is the pixel data processed. *See id.* at 22:26-32.

B. Williams (PCT Pub. No. WO00/63770A1)

48. Williams (Ex. 1006) discloses an apparatus and method for improving a graphics bus. It targets the efficiency of transmitting graphics commands from a CPU to a graphics subsystem, by distributing the transmission across several busses and reconstructing the original order of commands through the use of tags. *See* Ex. 1006 at Abstract. Specifically, Williams identified the speed of graphics data transmitted from a CPU to a graphics subsystem is limited by the bandwidth of and congestion at the peripheral bus. *See id.* at 3:27-4:5. Williams’s disclosure is directed at “increas[ing] the throughput of graphics commands that can be transferred between the system bridge and a graphics subsystem.” *Id.* at 8:4-10.

49. Williams, in the non-regrouping embodiment, uses a buffer to temporarily store graphics commands. *Id.* at 9:10-12. In this embodiment, the graphics commands can be transferred from the system bridge over bus lines via a sorting unit. *Id.* at 9:8-10. The graphics commands are delivered to the graphics subsystem in the same order as they are transferred from the CPU. *Id.* at 23:14-19 (providing that graphics commands are transferred, “according to the tags, such that the multiple graphics commands regain the specific order (that is, their original order)).

C. Whittaker (U.S. Patent No. 5,968,167)

50. Whittaker (Ex. 1007) discloses a generic multi-threaded “data processing management system” to “process multimedia data.” Ex. 1007 at Abstract. This “management system” consists of a “media control core” and “a number of data processing units.” *Id.* The “multimedia core identifies the data processing operation to be executed as well as the resources needed to execute that operation.” *Id.* The media control core “determines which operation has the highest priority” and passes that operation on “to one of the data processing units for execution.” *Id.*

51. Whittaker was designed to implement an array of independent vector pipeline processors, where each stage of the pipeline performed a fixed function under the control of the data processing core. “[The data processing] units consist of a processing pipeline (data pipeline 10) which will be made up of a number of processing elements such as multipliers, adders, shifters, etc., under the control of an associated data processing core 8 which runs a sequence of instructions to perform a data processing algorithm.” *Id.* at 4:27-33.

IX. EVALUATION OF PETITIONER’S PROPOSED GROUNDS

A. Ground 1: Stuttard and the Knowledge of a POSITA Does Not Render Claims 1-9 of the ’053 Patent Obvious

52. In my opinion, claims 1-9 are not obvious in view of Stuttard and the knowledge of a POSITA.

**1. Stuttard Does Not Disclose or Render Obvious the Claimed
“at least one memory device”**

53. Claims 1 and 5 require “at least one memory device comprising a first portion operative to store a plurality of pixel command threads and a second portion operative to store a plurality of vertex command threads.” Ex. 1001, Claims 1, 5. In my opinion, this claim limitation requires the storage of pixel command threads and vertex command threads in different portions of memory *at the same time*. See *supra* Section V.

54. Dr. Pedram identifies the cache unit 1024 as disclosing the claimed “at least one memory device . . .” of claim 1 and 5. Ex. 1004, ¶ 59. However, even Dr. Pedram admits that “Stuttard does not explicitly disclose that this cache unit 1024 stores a plurality of vertex command threads and a plurality of pixel command threads.” *Id.*, ¶ 62. Thus, Dr. Pedram relies on an alleged “obvious” modification to the memory device to store both types of command threads and to do so in separate “portions.” *Id.*, ¶¶ 62-64.

**a. Dr. Pedram’s Proposed Modification is Illogical
Given Stuttard’s Phased Architecture and SIMD
Design**

55. Dr. Pedram claims that it would be obvious to modify Stuttard to meet the claim limitation. In my opinion, there is no motivation for Dr. Pedram’s proposed modification, for two main reasons.

56. **First**, as explained above in Section VIII.A, and as Dr. Pedram agreed, Stuttard uses a phased system that operates first only on vertex threads and then only on pixel threads. Dr. Pedram testified at the ITC hearing:

Q. You will agree with me, at least, that in Stuttard's linear graphics pipeline, the process of pixel processing occurs after geometry processing; correct?

A. I know ***pixel processing occurs after vertex processing***. That, I know.

Ex. 2005 (ITC Hearing Day 4 Tr.) at 994:21-25. Thus, there is no reason to store vertex threads and pixel threads at the same time—much less the additional requirement dedicate one “portion” to pixel threads and another “portion” to vertex threads. Such resource allocation would waste the valuable, limited resources of a small cache memory to store threads that are not capable of being executed half the time. **Second**, Stuttard's system only operates on one instruction at a time, as explained above in Section VIII.A. Thus, there is no reason or even ***ability*** to modify Stuttard to be a non-phased system or to dedicate portions of memory to both vertex and pixel threads simultaneously.

57. The fact that Stuttard's cache unit 1024 would not be modified to store pixel and vertex command threads at the same time is further bolstered by Stuttard's SIMD architecture. Given Stuttard's SIMD design, it is incapable of processing both pixel threads and vertex threads at the same time. As Dr. Pedram admitted, it is well

known to a POSITA, that a SIMD processes only a single instruction at one time. Ex. 2005 at 995:20-22 (providing that SIMD means that the “[s]ame instruction sequence is processing different data items”).

58. In my opinion, it is not obvious to store pixel and vertex commands at the same time in different portions of memory “for ‘faster and convenient storage and retrieval functions, and thereby enhancing the logical layout of the thread processing system,’ taking ‘advantage of Stuttard’s multithreaded architecture that can more efficiently and speedily process graphics data.’” Ex. 1011 at 95-96. This is because those hypothetical advantages do not exist in a phased system with a single SIMD (that can only execute one instruction at a time).

59. In my opinion, these changes waste resources and create overhead for no benefit—a POSITA would recognize that it would waste resources for a phased system to store commands (or set aside storage space for commands and leave it empty) for a phase different from the current phase. For example, storing vertex commands in memory while processing pixel commands would reduce the memory available to process pixel commands during the pixel processing phase, which a POSITA would understand could lead to performance degradation not justified by any discernible benefit. That is, because Stuttard can only operate in vertex or pixel thread mode, it would be an illogical design choice to modify the memory device to

dedicate valuable resources to storing pixel threads when in vertex mode and vertex threads when in pixel mode, because they cannot be used at those respective times.

60. Furthermore, Stuttard is a phased system. *See* Section VIII.A.2 *supra*. Dr. Pedram agrees with me that it is phased. Ex. 1011 at 98. Although Stuttard provides that “[t]he thread scheduler, when running, recalculates which thread should be active,” the thread scheduler does not enable the cache memory to store pixel and vertex command threads at the same time. Ex. 1005 at 23:35-24:2. Hypothetically speaking, the thread manager could choose from pixel and vertex threads *if* both were stored at the same time (which they are not), and switch from one to the other as it chooses as described by Stuttard above. However, nothing in Stuttard indicates that the cache unit stores pixel and vertex command threads at the same time such that the thread manager could choose between them.

61. I disagree that the cache unit could be modified to store both type of threads at the same time. Modification of the memory to simultaneously store both type of command threads for a “faster and convenient storage and retrieval functions” is a benefit that does not exist in a phased system as explained above.

b. The Alleged AAPA Does Not Change The Analysis

62. Dr. Pedram’s opinion that the AAPA allegedly stored both types of threads does not provide a motivation for the proposed modification, as it would still conflict with Stuttard. Ex. 1004, ¶ 65. That is, the proposed modification to Stuttard

is illogical as explained above, even if it was generally “known” that one could store both types of commands, as Dr. Pedram alleges, because it is a bad design decision given the rest of Stuttard’s system.

63. Further, even if Stuttard could be combined with the alleged AAPA, which as explained above lacks any motivation, Dr. Pedram has failed to prove that the combination of Stuttard and the alleged AAPA discloses “at least one memory device” which stores pixel command threads and vertex command threads *at the same time*. Dr. Pedram asserts that the alleged AAPA discloses storing different types of command threads in different memory portions in a graphics processing system. Ex. 1004, ¶ 65. This is incorrect because none of the cited portions of the alleged AAPA disclose pixel and vertex command threads (or any command thread type at all), and most importantly, does not disclose storing them at the same time. See Ex. 1001 at 1:42-46. In fact, the alleged AAPA states that it has “specifically delineated ALU resource buffers and texture fetch resource buffers such that command threads must be *sequentially* provided between the various buffers.” Ex. 1001 at 1:65-2:1. This sequential provision between different buffers lends further support to the alleged AAPA’s nondisclosure of storing pixel and vertex command threads at the same time.

2. Stuttard Does Not Disclose or Render Obvious Storing Command Threads

64. Claims 1 and 5 of the '053 patent require storing “command threads.” As construed by the ALJ, “a complete thread must be stored at a given moment in time.” *See* Section V. It is insufficient to point to “portions of a complete thread, even if “the entire thread is stored over a period of time.” *Id.*

65. Stuttard does not disclose storing complete command threads. Dr. Pedram’s alleged memory device, cache unit 1024, only stores individual instructions, *i.e.*, commands, with no indication that those commands comprise complete command threads. *See, e.g.*, Ex. 1005 at 11:3-5 (“The thread manager 102 is shown in more detail in Figure 4, and comprises a cache memory unit 1024 for 5 storing instructions fetched for each thread”), 11:12-14, 23:3-6, 23:13-16. Dr. Pedram agreed that an “instruction would be a type of command,” *i.e.* not a command thread. Ex. 2008 (Pedram 1/4/23 Dep. Tr.) at 29:11-14. Therefore, at best, cache 1024 stores instructions that are subparts of a command thread, but as Petitioner argues in its claim construction, that is insufficient to store “command threads.” *See* Section V. Dr. Pedram is in agreement that Stuttard fails to disclose storing command threads: he testified that Stuttard does not clearly *ever* store a complete command thread in cache 1024 at any given moment in time:

Q So at any given moment in time --

A Yes, sir.

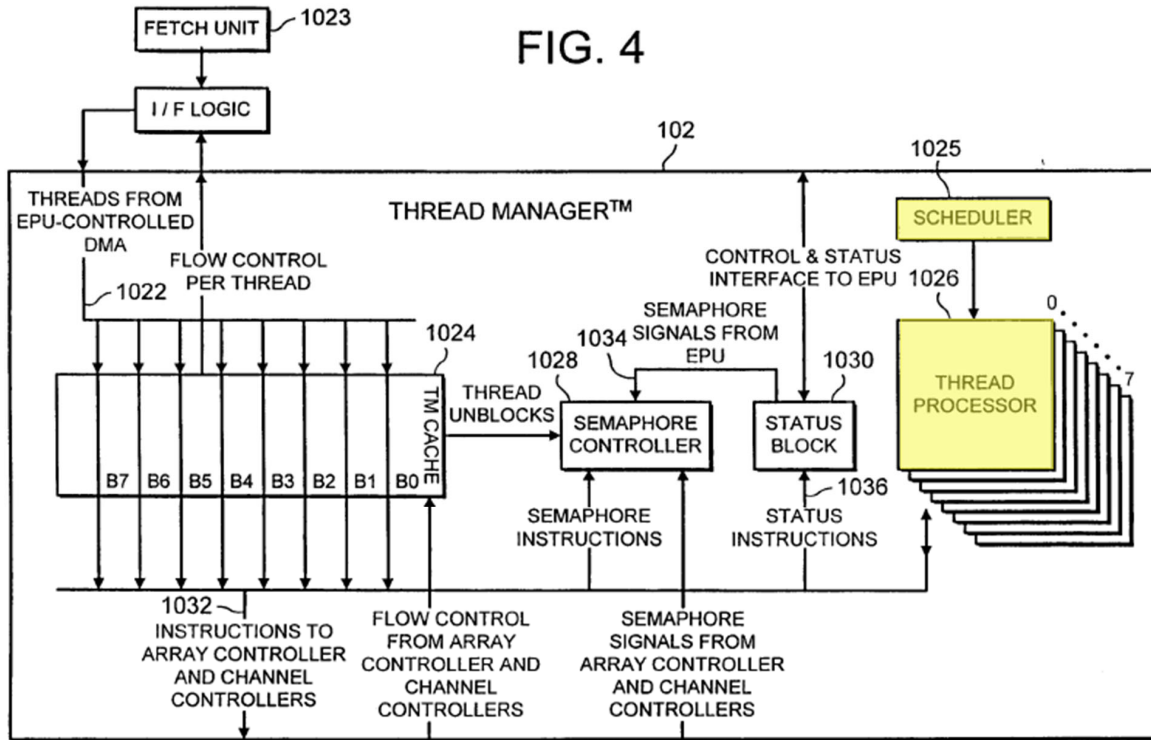
Q -- the cache unit 1024 might never store all of the instructions in a thread?

A May or may not.

Ex. 2008 at 183:6-10.

3. Stuttard's Alleged Arbiter Does Not Select Between a Plurality of Pixel and Vertex Command Threads

66. Claims 1 and 5 require “an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads.” Ex. 1001 at Claims 1, 5. Dr. Pedram opines that thread scheduler 1025 and thread processors 1026 together correspond to the claimed arbiter.” Ex. 1004, ¶ 68.



Ex. 1005 at Fig. 4 (annotated).

67. However, as explained above in Section IX.A.1, for the “memory device” limitation, there is no point in time when Stuttard’s cache unit 1024 stores both pluralities of pixel command threads and vertex command threads, given Stuttard’s phased operation and SIMD design. This is true even if the above claim construction regarding the memory device is incorrect. Given Stuttard’s express disclosure of a phased system that executes a single instruction at a time, there is no opportunity for the arbiter to select between at least two pixel command threads and at least two vertex command threads.

68. Stuttard's thread manager also does not render the claimed memory device obvious. In my opinion, when Stuttard's phased system is processing vertex command threads, only vertex command threads exist in cache unit 1024—pixel command threads do not—and vice versa when the system is processing pixel command threads. Thus, thread manager cannot “select a command thread from either of the plurality of pixel command threads *and* the plurality of vertex command threads.”

69. The alleged AAPA (Fig. 1 of the '053 Patent) does not solve Stuttard's deficiencies because the structure of Fig. 1 is fundamentally inconsistent with Stuttard's phased operation and SIMD architecture that operates on one instruction at a time in a particular phase. Further, the AAPA is insufficient, because even if the “ALU resource” must be used for vertex commands and the texture fetch resource must be used for pixel commands, the system uses a FIFO – first in first out – for each of the two buffers the arbiter has access to. Ex. 1001 at 1:35-36, Fig. 1. Thus, the arbiter cannot select among the required four items – at least two vertex commands and at least two pixel commands; it only has access to the first item coming out of the buffer and does not select among commands of the same type (*i.e.*, the required “pluralities” component of the claim limitation). Dr. Pedram has not explained how Stuttard's architecture would be modified to fit the architecture of the AAPA, or vice versa.

B. Ground 2: Stuttard, Williams, and the Knowledge of a POSITA Does Not Render Claims 1-9 of the '053 Patent Obvious

70. In my opinion, claims 1-9 are not obvious in view of Stuttard, Williams, and the knowledge of a POSITA.

1. Williams Does Not Disclose Or Render Obvious A Memory Device That Stores A Plurality Of Pixel Command Threads And Vertex Command Threads At The Same Time

71. As a preliminary matter, just as with the AAPA, explained above in Section IX.A.1.b, even if Williams taught the claimed memory device of this limitation (which it does not, as explained more fully below) proposing to modify Stuttard with Williams fails for the same reasons as with Stuttard alone and Stuttard with the alleged AAPA, as described above: it proposes a modification to Stuttard that contradicts Stuttard's fundamental design and has no motivation.

a. Williams Does Not Disclose or Render Obvious Storing Command Threads

72. Williams does not disclose the claimed memory device for multiple reasons. Williams does not store "command threads" at all, let alone both pixel and vertex command threads at the same time in separate portions of memory, as required by claims 1 and 5 of the '053 Patent. A "command thread" is a "sequence of commands applicable to the corresponding element," where the "corresponding element" refers either to vertex data or pixel data. Pet. at 14. That is, there must be some relationship between multiple instructions that are sequenced together, and

they must be applicable to particular vertex data or pixel data, to constitute a thread. But nothing in Williams describes storing such a “sequence” of complete pixel and vertex command threads. For example, Williams at Table 1 demonstrates its invention using “exemplary OpenGL pseudo-code graphics commands ... selected from a test program that draws a random series of connected triangle strips.” Ex. 1006 at 12:20-22. These commands set OpenGL state variables, such as the rendering mode, and send vertices (including attributes) to the graphics card. These exemplary commands in Table 1 do not include any vertex command threads executed by a vertex shader, nor any pixel command threads executed by a pixel shader. (In fact, Williams was filed well before user-programmable vertex shaders and pixel shaders even existed in a GPU.)

73. Dr. Pedram identifies the system memory and/or the input buffer of Williams’s Fig. 4 embodiment as disclosing the claimed “at least one memory device ... limitation.” Ex. 1004, ¶¶ 96-97.

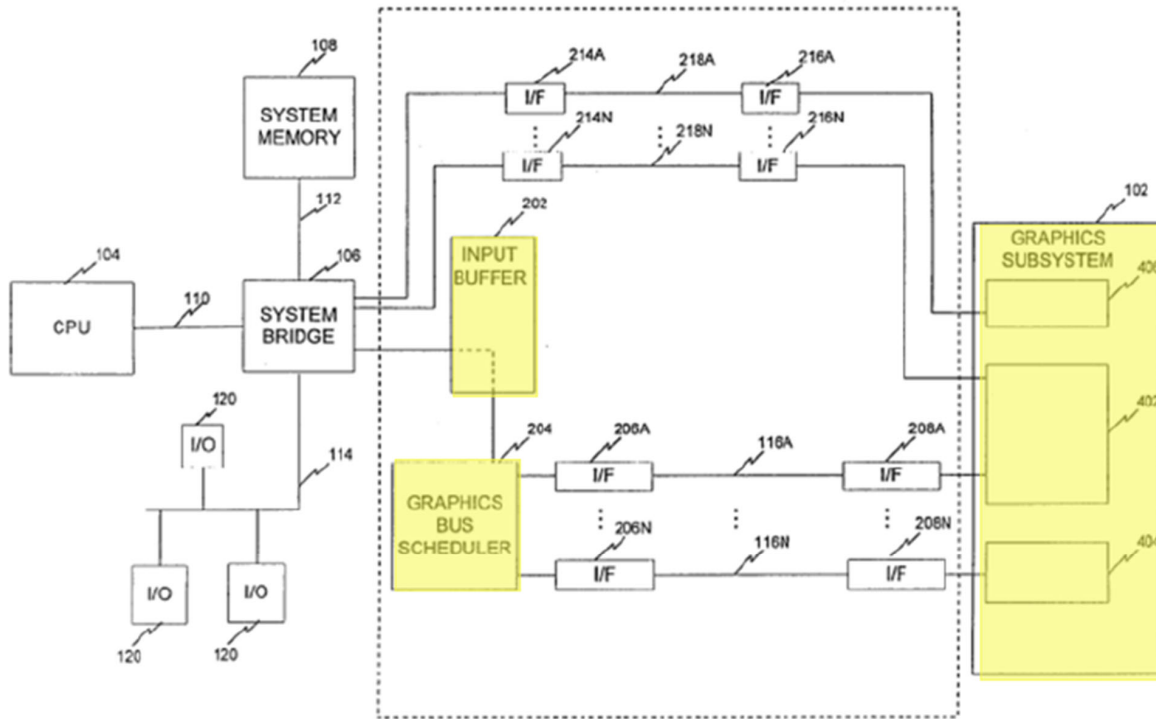


FIG. 4
NON-REGROUPING EMBODIMENT

Ex. 1006 at Fig. 4. Rather, Williams explains that individual “commands” are put into the input buffer 202, without explaining any relationship between multiple commands. Ex. 1006 at 20:14-26. Then “GBS 204 assigns/sorts graphics commands according to their type” and puts them on the appropriate bus. *Id.* Specifically, the “graphics commands are not reordered,” *i.e.*, are not grouped into a “thread.” *Id.*

74. Williams uses multiple busses to transmit an ordered sequence of graphics commands from the CPU to the GPU. While multiple busses are used to transmit these commands from CPU to GPU, the order they are processed by the

GPU is the same order that they provided from the CPU. This is a clear demonstration of ordinary (single-threaded) execution, and a POSITA would not have understood such execution to demonstrate multi-threaded execution, which would change the order of execution of commands depending on e.g. resource availability and/or priority.

75. Further, Williams discloses an invention for graphics *bus* technology, not graphics *processing* technology. Williams points to the graphics pipeline shown in Fig. 7 specifically for the non-regrouping embodiment in Fig. 4. *See id.* at 25:15-27. Williams also states that Fig. 7 is an example graphics processing pipeline that “can include but is not limited to” OpenGL. *Id.* at 23:23-24:4. Graphics pipelines, including OpenGL, are state machines, meaning that the CPU sends commands to them to set up the pipeline, then the CPU sends primitives (triangles) through the pipeline for processing. These state machines are deterministic and order dependent, meaning they *must* maintain the order of the state variables and primitives. For example, the CPU could set the color to red, then send three triangles, then set the color to blue, and send three more triangles. If the order gets switched, then you could get e.g. two red triangles and four blue triangles. The same is true for processing geometry (e.g., triangles) and texture images (which would be like the color assigned to the triangles).

76. Since Williams discloses a graphics bus and not a graphics processor, it makes assumptions about the graphics pipeline in Fig. 7 – namely, that it would maintain the order of commands, even if data arrived to it across several independent busses. So even though Williams is sending vertex data across one bus and pixel data across another, the graphics pipeline in Fig. 7 and its underlying graphics processor *must* still process the command in the same order they were sent by the CPU to ensure the graphics state remains consistent and set properly for each primitive processed.

b. Williams Does Not Disclose or Render Obvious A First Portion Operative To Store A Plurality Of Pixel Command Threads And A Second Portion Operative To Store A Plurality Of Vertex Command Threads

77. Even if Williams were to store command threads, there is no indication of the claimed dedicated “first portion” and a “second portion” of memory, each portion respectively storing pixel and vertex command threads. It simply discloses a buffer for temporary storage of commands. Ex. 1006 at 11:20-24 (“Input buffer 202 . . . temporarily stores graphics commands that are to be transferred from system bridge 106 to graphics subsystem 102.”), 16:28-30, 17:26-27. Dr. Pedram concedes by arguing it would have been obvious to dedicate these first and second portions within memory to “accomplish Williams’ goal of greater efficiency of command fetching and transfer.” Ex. 1004, ¶ 99.

78. Dr. Pedram’s alleged motivation for its proposed modification does not actually benefit Williams or Stuttard in any way, and would not be carried out by a POSITA. Dr. Pedram opines that “it was well-known that commands stored in consecutive addresses can be read/written with fewer instructions than the non-consecutive addresses because reading non-consecutive addresses involves address jumps.” *Id.* But those hypothetical benefits have no relevance to Williams. In Williams, the Graphics Bus Scheduler “sorts the graphics commands according to the type of command” and put them on the appropriate bus line. Ex. 1006 at 20:14-26. That is, the GBS receives the instructions one by one from the input buffer and puts them onto the appropriate bus line. Ex. 1006 at 20:25-26 (“GBS 204 receives multiple graphics commands directly from system bridge 106, or from input buffer 202.”). There are no “address jumps” in Williams, because the GBS does not read from memory addresses to obtain the commands; it simply receives them from buffer as they exit, one after the other. The hypothetical benefit of eliminating address jumps and thus using fewer read instructions has no application to Williams, because there are no address jumps or read instructions, and so do not provide a motivation to modify Williams.

79. In fact, modifying Williams to require reading particular memory addresses in order to retrieve commands would substantially slow the system down. Making the proposed change so that given types of commands were in

predetermined memory locations would also render redundant the GBS unit, whose job it is to sort the commands by type based on the randomly-ordered assortment that exit the input buffer. Dr. Pedram has failed to show that his hypothetical proposed system is more “efficient” than the system Williams uses, or any reason a POSITA would make the proposed change.

80. Dr. Pedram’s alternative theory, regarding Figure 6 of Williams, fails for the same reasons. Ex. 1004, ¶ 100. Dr. Pedram identifies Williams’ input buffer and output buffer of Fig. 6 (reproduced below) as disclosing the “at least one memory device” claim limitation. *Id.* Dr. Pedram asserts that the input buffer stores pixel commands and the output buffer stores vertex commands, thus meeting the claim requirements. *Id.* This argument suffers from all of the same arguments above, *e.g.*, this disclosure merely stores commands and not command threads. Additionally, there is no indication that they are stored “at the same time,” or in “separate portions” as some relate to input, before the buses, and some relate to output, after the buses.

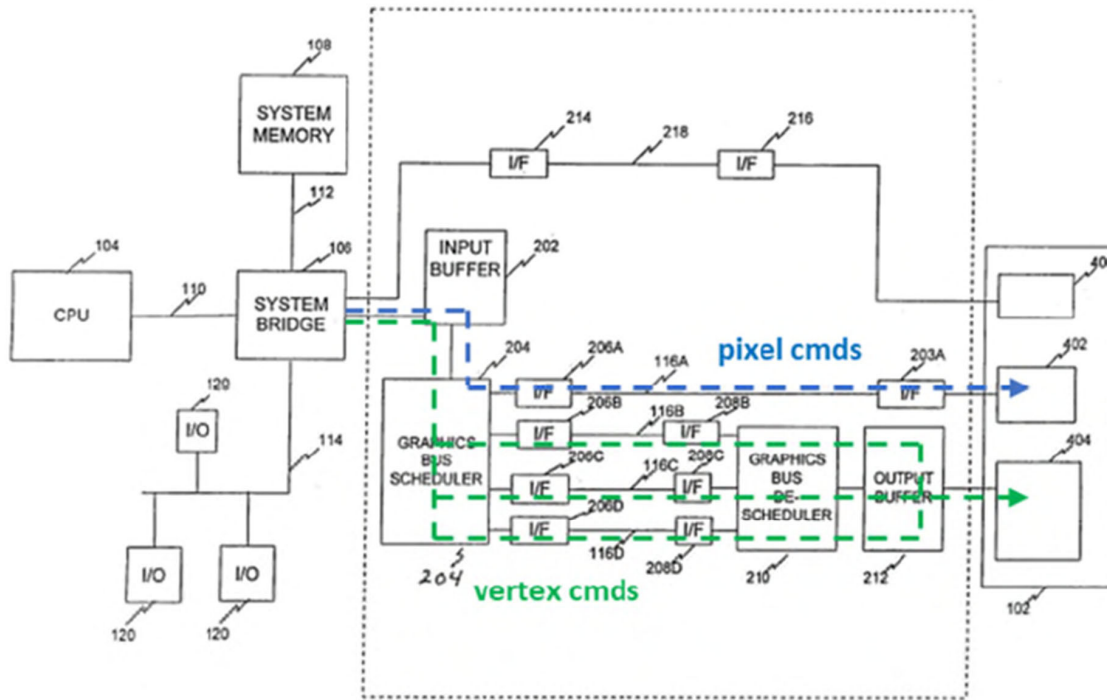


FIG. 6

Ex. 1006 at Fig. 6 (annotated by Dr. Pedram); Ex. 1004, ¶ 100. Moreover, these two buffers cannot satisfy the claimed “memory device” as they are not “coupled to” anything that could be an “arbiter,” which is required in the claims to “select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads based on relative priorities of the plurality of pixel command threads and the plurality of vertex command threads.” Ex. 1001 at claim 1. Dr. Pedram’s limited analysis contains no allegations regarding how any alleged “arbiter” would connect to both an input buffer and an output buffer, which serve entirely different purposes, such that it could select between the types of command threads, much less how a system would be designed such that an arbiter

that could select from the above differently-situated input and output buffers would then be coupled to “a plurality of command processing engines . . . each operable to receive and process the command thread.” Ex. 1001 at claim 5. These two non-analogous buffers simply make no sense as a “memory device” for storing pixel and vertex command threads for execution via an arbiter transmission to command processing engines.

2. There Is No Motivation to Combine Williams with Stuttard

81. Even if Williams taught or rendered obvious the claimed memory device, which it does not as explained above, a POSITA would not be motivated to combine Williams’s system memory, input buffer, or output buffer with, or as a replacement for, Stuttard’s cache unit 1024, as Dr. Pedram alleges. Dr. Pedram’s motivation to combine argument rests on the two references being both directed to graphics processing, without taking into consideration the nuances of each. *See* Ex. 1004, ¶ 102 (asserting that both references “have overlapping components, such as a CPU in a host system and a graphics subsystem,” etc.).

82. First, the functionality of Williams’s system memory, input buffer, and output buffer is already redundant of Stuttard’s local memory and system memory devices. A POSITA would not replace Stuttard’s cache unit with William’s system memory where the functionality of these devices is redundant of functionality already found elsewhere in Stuttard’s system. For example, Williams’ input buffer,

which is used “as an interface between system bridge 106 and GBS 204.” Ex. 1006 at 18:21-22. This functionality is redundant of the functionality already present in Stuttard’s EPU and/or local memory, which also acts as a bridge between the host system and the graphics system and has storage for command threads. Ex. 1005 at 7:1-9, 11:12-14, 23:1-16 (“storing instructions fetched . . . from local external memory 103 or from the EPU 8 by the fetch unit 1023”), Fig. 2. Finally, replacing Stuttard’s high speed cache with Williams’ lower speed buffer or system memory would render Stuttard inoperable for its intended purpose of enabling high speed instruction execution. *See generally, e.g.*, Ex. 2007.

83. Moreover, Dr. Pedram’s purported motivation to combine speculates that it makes sense to store the different types of commands in different areas of memory, but there is no reason to do that given the role of the GBS within Williams’ architecture. Specifically, William provides that upon system memory or input buffer, commands are sorted by GBS onto an appropriate bus. *E.g.*, Ex. 1006 at 11:20-12:2 (providing that GBS 204 “assigns each of the multiple graphics commands” that are “stored within input buffer 202” “to one of the plurality of busses”). If system memory 108 or input buffer 202 already stored different graphics commands in separate portions (which they do not), then the functionality of GBS 204, which “assigns and/or sorts the graphics commands according to the type of command,” would be redundant because the commands would have already been

sorted by type in system memory 108 or input buffer 202. *Id.* at 19:4-5. Nor would there be a need for GBS 204 to assign the graphics commands to different buses based on the type of command. *Id.* Having this kind of redundancy would not achieve Williams' purported goal of greater efficiency of command fetching and transfer.

84. For the same reason, Dr. Pedram's purported hypothetical benefits from grouping similar data at near address spaces in memory also fails. Ex. 1004, ¶ 99. Such batching may be useful if a system is capable transferring a batch of data, but that is not what Williams teaches. *See* Ex. 1006 at 11:20-12:2. The graphics commands exit Williams' input buffer, and get sorted by the GBS on the way out. *Id.*; *see also id.* at 18:26-19:11 ("GBS 204 receives multiple graphics commands directly from system bridge 106, or from input buffer 202," and "assigns each of the multiple graphics commands to one of the plurality of busses 116A-116N."). That is, each graphics command is individually assigned by GBS 204.

85. Moreover, even if the system memory and/or input buffer stored a plurality of vertex/pixel command threads in first and second portions, a person of ordinary skill in the art would not be motivated to modify Stuttard's cache unit 1024 to incorporate this structure. As previously discussed, Stuttard discloses a phased SIMD structure, which is only capable of executing a single instruction at a time. As Stuttard's system is incapable of executing pixel threads until it completes all vertex

processing, much less at the same time as executing vertex threads, storing both of a plurality of pixel threads and vertex threads in separate first and second portions of the instruction cache 1024 would waste valuable storage resources to store threads that are incapable of being executed. As a result of this inefficient use of a caching strategy, the modified version of Stuttard would encounter a greater number of cache misses, making Stuttard's overall system performance worse.

C. Ground 3: Williams, Whittaker, and the Knowledge of a POSITA Does Not Render Claims 1-9 of the '053 Patent Obvious

86. In my opinion, claims 1-9 are not obvious in view of Williams, Whittaker, and the knowledge of a POSITA.

1. Williams Does Not Disclose or Render Obvious Storing Command Threads

87. As explained above in connection with Ground 2, Williams does not disclose storing pixel and vertex command *threads*, as required by all challenged claims. *See* Section IX.B.1. Additionally, as explained above in Section IX.B.1, Williams also does not disclose storing any graphic commands in separate portions, at the same time.

2. Williams Does Not Disclose Or Render Obvious An Arbiter That Selects A Command Thread

88. Both independent claims 1 and 5 of the '053 patent require "an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command

threads.” Ex. 1001 at Claims 1, 5. Dr. Pedram opines that GBS 204 operates as an arbiter. Ex. 1004, ¶ 133.

89. However, Williams’s GBS does not *select* graphics commands (or anything). Rather, as Williams provides, GBS “*receives* multiple graphics commands directly from system bridge 106, or from input buffer 202,” and “assigns each of the multiple graphics commands to one of the plurality of busses 116A-116N.” Ex. 1006 at 18:26-28 (emphasis added). As explained above in Ground 2, the GBS merely receives the commands, one by one, coming out of the input buffer and sorts them; it does not select commands. Moreover, the claim requires selecting between at least two pixel command threads and two vertex command threads, by virtue of requiring a “plurality” of *each* type, *i.e.*, at least two of each type, which is certainly not taught by Williams or Whittaker (or the combination).

3. Williams Does Not Disclose Or Render Obvious Command Processing Engines

90. Claim 5 of the ’053 Patent requires “a plurality of command processing engines, coupled to the arbiter, each operable to receive and process the command thread.” As explained above, Williams in view of Whittaker does not disclose or render obvious pixel command threads or vertex command threads. Therefore, Ground 3 does not disclose a command processing engine to receive (and process) the command.

91. Further, for claim 5, Dr. Pedram cites to the same evidence as claim 2, claiming that “[e]lement 5[c] is identical to claim 2 with multiple command processing engines.” Ex. 1004, ¶ 147. However, claim 5 differs from claim 2 not only because it requires more than one command processing engine, but because *it also requires* each command processing engine is operable to receive and process the command thread. Ex. 1001 at Claims 2, 5. Meanwhile, claim 2 differs from claim 5 because it requires the arbiter is operable to provide the command thread to the command processing engine. *Id.* Therefore, Dr. Pedram has failed to explain how the plurality of command processing engines receive and process the command thread, which is specific to claim 5.

92. Dr. Pedram identifies “functional components” 402 and 404 in Williams as the claimed “command processing engines” because they perform geometry and pixel operations. Ex. 1004, ¶ 140. However, these are not “command processing engines” at least because Dr. Pedram has made no attempt to demonstrate they are coupled to the GBS 204, the alleged arbiter, as they must be per the claims. Nor do these functional components receive both pixel and vertex commands from the GBS 204 (as required by claim 2).

93. Nor are they both operable to process both pixel and vertex commands (as required by claim 5). Williams discloses that functional components 402 and 404 “perform geometry and pixel operations, *respectively*,” meaning that each

functional component only performs (and thus receives) one operation, but not the other. Ex. 1006 at 22:5-7 (emphasis added). Similarly, Williams also discloses that the graphics commands can be “transferred to the appropriate component/stage of a graphics pipeline 700.” *Id.* at 22:7-9; *see also id.* at 25:15-23. Thus, there are only dedicated, type-specific components in Williams alleged to be the claimed command processing engines. Meanwhile, the command processing engines of claim 5 are capable of receiving and processing both the pixel command thread and vertex command thread, depending on which command thread is selected by the arbiter. That is, the command processing engines cannot be type-specific. Dr. Pedram agreed that claim 5 excludes type-specific processing engines, that is, command processing engines which processed only pixel or only vertex command threads. Ex. 2005 at 988:8-19. He agreed that each of the command processing engines must be operative to process whichever thread is provided to it by the claimed arbiter, be it a pixel command thread or a vertex command thread. *Id.*

4. Neither Williams Nor Whittaker Discloses Or Render Obvious An Arbiter That Selects Command Threads Based On Relative Priorities Of The Command Threads

94. Claim 1 of the '053 patent requires “an arbiter, coupled to the at least one memory device, operable to select a command thread from either of the plurality of pixel command threads and the plurality of vertex command threads based on relative priorities of the plurality of pixel command threads and the plurality of

vertex command threads.” Ex. 1001 at Claim 1. Dr. Pedram does not opine that Williams discloses selection “based on relative priorities” as required by claim 1. Ex. 1004, ¶ 133. For this, Dr. Pedram relies on Whittaker’s “MCC 2.” *Id.* Whittaker, however, also does not disclose pixel command threads or vertex command threads at all, or even different types of command threads, let alone storing in separate portions both pixel and vertex command threads at the same time and then selecting from pluralities of each type of thread. Therefore, even if MCC 2 selects threads based upon thread priority, nothing in the combination of Williams with Whittaker discloses an arbiter that selects between at least two pixel command threads and at least two vertex command threads.

95. Further, a POSITA would not have been motivated to make the modification to Williams from Whittaker that Dr. Pedram suggests for this element. As explained further below, Williams is directed to overcoming deficiencies of the peripheral bus and increasing the throughput of the graphics data to the graphics subsystem, whereas Whittaker is concerned with the efficient processing of all multimedia data by evaluating resource availability and thread priority.

96. Specifically, the MCC 2 of Whittaker is a fine-grained multithreading processing unit which is in control of all the scheduling and processing of media data. *See* Ex. 1007 at 3:45-49, 4:35-43. In order to select tasks based on task priority, the MCC 2 checks, on every clock cycle, “which of the possible operations

it could perform have all the resources available for those tasks to be executed and, of those, which has the highest priority.” *Id.* at 3:56-4:24. The resource checking and priority is illustrated in Figure 6, which explains that for each thread, “global status information is received from the necessary data banks, the necessary address banks, routing control data from the control unit, control status information from control unit 16, and execution dependency data from other processes on which a particular thread is dependent.” *Id.* at 8:17-23, Fig. 6. “All of this information is sent to a resource checker 81 which combines it with data from 10 ports, the various pipeline data bank status, and the status of various data processing units, . . . for each possible thread.” *Id.* at 8:23-27.

97. Unlike the MCC 2, Williams’ GBS 204 does not control the components of the graphics subsystem which processes the graphics commands. *See generally* Ex. 1006. Furthermore, in order for GBS 204 to select a thread based the priority of the thread as disclosed in Whittaker, it would also require GBS 204 to perform all the checks and receive all the information disclosed in Whittaker necessary to determine the thread priority. *See* Ex. 1007 at 3:56-4:24, 8:17-35. Dr. Pedram has not adequately explained how a POSITA would understand all the required aspects of MCC 2 to select based on relative priorities would fit into or with GBS 204.

5. There is No Motivation to Combine Williams with Whittaker, Nor Any Reasonable Expectation of Success

98. A POSITA would not have been motivated to combine Williams and Whittaker. Williams targeted high-performance graphics through bus optimization, seeking to “increase[] the throughput of graphics commands transferred between a system bridge and a graphics subsystem.” Ex. 1006 at Abstract. In contrast, Whittaker sought a cost-saving general multimedia processor “which processes all multimedia data types in a manner that minimises system costs and provides for future developments in multimedia and the related industry standards,” by prioritizing which threads are being processed. Ex. 1007 at Abstract, 2:4-6. Thus, Williams and Whittaker are aimed at optimizing different components of a graphics processing system, where Williams is directed to the transfer of graphics commands to a GPU, and Whittaker is directed to the prioritization of threads being executed.

99. Further, a POSITA would not be motivated to combine Whittaker with Williams, because they operate at completely different levels of process analysis. Whittaker’s MCC 2 analyzes threads being processed based on their media needs, *e.g.*, sound, 3-D graphics, 2-D graphics, MPEG video, whereas Williams analyzes only the activity of moving 3-D graphics commands from the CPU to the GPU. *E.g.*, Ex. 1007 at Abstract, 3:36-44, 56-14, 6:42-49, Fig. 1; Ex. 1006 at Abstract. Further, Williams provides assigning graphics commands (*e.g.*, an OpenGL command) to

different buffers, and it would not be obvious to a POSITA, even in light of Whittaker, how to assign such priorities to individual OpenGL commands.

100. A POSITA would not have held a reasonable expectation of success from the combination of Williams and Whittaker. Williams discloses a method for scheduling the transmission of graphics commands across multiple busses by assigning them to different busses based on their type, or order (via first tagging to indicate the specific order of the commands). Ex. 1006 at 18:14-20, 23:3-21. This approach of Williams conflicts with Whittaker which is designed to be able to ignore the specifics of the media type and instead examines lower-level resource needs, using a single Media Control Core (MCC) that bottlenecks the connection between the data sources and the processing cores. *See, e.g.*, Ex. 1007 at Abstract. This proposed combination would not have provided the throughput sought by Williams, nor the generality sought by Whittaker.

X. CONCLUSION

101. In signing this declaration, I recognize that it will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place in the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

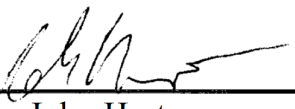
102. The statements and opinions in this declaration comprise and are based upon my review, analysis and understanding of the source materials that I have reviewed as discussed herein, based on the disclosure and statements made within such source materials.

103. I reserve the right to supplement my opinions in the future to respond to any arguments that the Petitioner raises and to take into account new information as it becomes available to me.

104. I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

July 27, 2023

Respectfully Submitted,



Dr. John Hart

EXHIBIT A

Curriculum Vitæ

John C. Hart

Director of Professional Education
and
Professor of Computer Science
University of Illinois at Urbana Champaign

(217) 333-8740 – jch@illinois.edu – <http://graphics.cs.illinois.edu/~jch>

∞ Experience ∞

Education

- **Ph.D.** Electrical Engineering and Computer Science Dept., University of Illinois at Chicago, 1991. Thesis title: "Computer Display of Linear Fractal Surfaces." Advisor: Thomas A. DeFanti.
- **M.S.** EECS Dept., University of Illinois at Chicago, 1989. Thesis title: "Image Space Algorithms for Visualizing Quaternion Julia Sets." Advisor: Thomas A. DeFanti.
- **B.S.** College of Liberal Arts and Science, Aurora University, 1987. Major: Computer Science.

Academic Positions

- **Director, Office of Professional Education**, University of Illinois at Urbana-Champaign, 2021-.
- **Director of Online Programs**, Dept. of Computer Science, 2018-2021.
- **Director**, Master of Computer Science in Data Science program, 2016-2017.
- **Executive Associate Dean**, Graduate College, University of Illinois, 2015-2021.
- **Associate Dean**, Graduate College, University of Illinois, 2014-2015.
- **Director for Graduate Studies**, Dept. of Computer Science, 2013.
- **Full Professor**, Dept. of Computer Science, University of Illinois, 2006-.
- **Associate Professor**, Dept. of Computer Science, University of Illinois, 2000-2006.
- **Associate Professor**, School of Electrical Engineering and Computer Science, Washington State University, 1998-2000.
- **Assistant Professor**, School of Electrical Engineering and Computer Science, Washington State University, 1992-1998.
- **Postdoctoral Research Associate**, Electronic Visualization Laboratory, University of Illinois at Chicago, and National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, 1991-1992.
- **Research Assistant**, Electronic Visualization Laboratory, Electrical Engineering and Computer Science Dept., University of Illinois at Chicago, 1988-1991.

- **Teaching Assistant.** Analog & digital circuit design courses, EECS Dept. University of Illinois at Chicago, 1987-1988.

Awards and Honors

- **The Excellence in Online & Distance Teaching Award.** University of Illinois at Urbana-Champaign, 2021.
- **Fellow.** Academic Leadership Program, Big Ten Academic Alliance, 2016-2017.
- **Outstanding Advisor.** College of Engineering, 2010 and 2015.
- **Winner.** 2004 Fantasy Graphics League.
- **Listed.** *International Who's Who of Professionals*, 1997.
- **Champion.** SIGGRAPH Bowl II, SIGGRAPH 94. Captain of the EVL Alumni Team.
- **NSF Research Initiation Award.** "Modeling, Rendering and Animation of Implicit Surfaces," to support the generalization of methods developed for the visualization of fractal models to the more common implicit surfaces used in CAGD and entertainment, 1993.
- **First Runner-Up.** Truevision Videographics Competition, SIGGRAPH 90.
- **Graduate Fellowship.** Graduate College, University of Illinois at Chicago, 1990-1991.

Industry Consulting

- **Intrinsic Medical Imaging LLC**, Bloomfield Hills, MI. Consultant, 2012-13. Mesh construction and simulation of coronary arterial blood flow.
- **Pratt & Whitney**, Hartford, CT. Consultant, 2011. Flux estimation in a lensed system.
- **Science Applications International Corp. (SAIC)**, Champaign, IL. Consultant, 2008. Ray-NURBS intersection.
- **Adobe Systems, Inc.**, Seattle, WA. Visiting Researcher, 2007. Project: Rendering meshed objects as stylized vector art.
- **The Teaching Company**, Chantilly, VA. Consultant, 2003. Produced custom educational video elements on the platonic solids.
- **Evans & Sutherland Computer Corp.**, Salt Lake City, Utah. Consultant, 1999-2000. Project: Design of the Evans & Sutherland Multi-Texturing Language.
- **Blue Sky | VIFX, Inc.**, Culver City, California. Consultant, 1998. Project: development of a custom software plug-in for the Houdini procedural animation package to polygonize a feature film villain modeled as a complex implicit surface.
- **Evans & Sutherland Computer Corp.**, (group formerly known as Silicon Reality, Inc.) Federal Way, Washington. Consultant, 1997-8. Project: design of graphics hardware to support antialiased procedural solid texturing.
- **Kleiser-Walczak Construction Company (a visual effects production company)**. Lennox, Massachusetts. Consultant, 1992-1993. Project: development of a new fractal-based video transition effect for an attraction at the Luxor Hotel, Las Vegas.
- **AT&T Pixel Machines.** AT&T Bell Laboratories, Holmdel, New Jersey. Summer Intern, 1990.
- **IBM T.J. Watson Research Center.** Hawthorne, New York. Summer Intern, 1989.

Expert Testimony

(Only includes disclosed engagements.)

- **LG Electronics, Inc.** v. TCL. US District Court, Eastern District of Texas, Marshall Division. C.A. No. 2:22-cv-122-JRG. Retained by Baker Botts, LLP. Authored expert decl. on claim construction, 4/18/23. Settled.
- **Advanced Micro Devices (AMD)** v. RealTek, TCL. US International Trade Commission Case #337-TA-1318. Retained by Mintz, Levin, Cohn, Ferris, Glovsky and Popeo PC. Testified at Markman Hearing 9/6/22. Ongoing,
- **Stratos Audio.** IPR's and trials regarding Patents 8,200,203, 8,688,028 and 8,903,307. Retained by White & Case LLP. Dec. 2021-ongoing. Deposed 3/31/22, 7/8/22, 7/12/22 and 7/20/22.
- **Infernal Technologies** v. Sony Interactive Entertainment, US District Court, Eastern District of Texas Marshall Division, C.A. No. 2:19-cv-248-JRG. Retained by Beuther, Joe & Counselors. Authored expert reports on infringement (Aug. 2020) and validity (Sep. 2020). Testified in jury trial Oct. 2021.
- **Intellectual Pixels**, Inter Partes Review for Patent 8,667,093. Retained by Armond Wilson. Mar. 2021 – June 2022. Authored declaration. Deposed 18 June 2021.
- **CorePhotonics**, Inter Partes Reviews for Patents 10,225,479 and 10,230,898. Retained by Russ, August & Kabat. Jan. 2021 – ongoing. Authored declarations. Deposed 29 Apr. 2021 and 21 May 2021.
- **LG Electronics Inc.**, US District Court, Central District of California, Western Division. Case No. 2:19-cv-09474-JAK. Retained by Sidley Austin Jan. 2020. Authored Sep 2020 expert report on claim construction.
- **Align Technology, Inc.**, US District Court for the District of Delaware, C.A. No. 17-1647-LPS-CJB. Retained by Morrison Foerester Mar. 2019. Authored infringement report and reply report on validity. Deposed 19 June 2020. Settled Feb. 2022.
- **Adsync Technologies**, Middle District of Florida, Orlando Division, Case 6:18-cv-1176-Orl-40TBS. Retained by Akerman Jun. 2019. Declaration in defense of Amended Complaint from Adacel and in support of counter complaint from Adsync. Deposed 24 Jan 2020. Settled Jan. 2021.
- **ZiiLabs** (subsidiary of Creative Technology), Certain Graphics Processors and Products Containing the Same, US International Trade Commission Case #337-TA-1099. Retained by Pepper Hamilton Sep. 2017. Deposed 12 Nov. 2018.
- **Samsung.** - FotoNation Ltd. et al. v. Samsung Electronics. Co., Ltd., et al., No. 2:17-cv-00669 (E.D. Tex.). Retained by O'Melveny & Myers Oct. 2017.
- **Samsung**, Inter Partes Reviews for of patents filed by Image Processing Technologies LLC: 6,717,518, 6,959,293, 7,650,015, 8,805,001, 8,983,134 and 8,989,445. Retained by O'Melveny & Myers Oct. 2016. Deposed Dec. 2017.
- **ZiiLabs** (subsidiary of Creative Technology), Certain Graphics Processors, DDR Memory Controllers, and Products Containing the Same, US International Trade Commission Case #337-TA-1037, for Pepper Hamilton LLP, Sep. 2016 – July 2017. Testified to the International Trade Commission for Markman hearing.
- **Advanced Silicon Technology**, Certain Computing or Graphics Systems, Components Thereof, and Vehicles Containing Same, US International Trade Commission Case #337-TA-984, for Mintz, Levin, Cohn, Ferris, Glovsky and Popeo PC, Aug. 2015 – Aug. 2016. Testified to the International Trade Commission for Markman hearing and authored validity report. Written declarations submitted for IPR2016-00894, -00897 and 01060).

- **ZiiLabs** (subsidiary of Creative Technology), Eastern District of Texas, Marshall Division, Case 2:14-cv-00203, for Heim, Payne & Chorush LLP and Susman Godfrey LLP. Sep. 2014 – Dec. 2015. Prepared invalidity rebuttal report for four patents and damages report for nine patents, all on graphics hardware. Deposed over two sequential days on both reports.
- **Graphics Property Holdings** (formerly SGI), Certain Consumer Electronics with Display and Processing Capabilities, US International Trade Commission Case #337-TA-884, for Mintz, Levin, Cohn, Ferris, Glovsky and Popeo . Graphics hardware expert on floating point rasterization, Authored report on patent validity. Deposed Feb. 2014. Testified to the International Trade Commission Apr. 2014.
- **Microsoft Xbox**, Illinois Northern District Court (Eastern Division) Case #08-C-393, for Sidley Austin LLP. Dec. 2009 – May 2010. Prepared non-infringement rebuttal report and invalidity report.
- **NVIDIA & AMD (ATI)**, California Northern District Court Case # 3:07-cv-00302, for Cooley Godward Kronish LLP. July 2008 – Sep. 2008. Provided expertise on graphics hardware and insight into the industry-wide process for the deployment of innovative features.
- **Broadcom**, California Southern District Court Case #06-CV-0660, for McAndrews, Held & Malloy, Ltd, Dec. 2006 – Mar. 2007.

∞ Research Funding ∞

From Industry

- **Caterpillar**, through the **Digital Manufacturing & Design Innovation Institute**. \$410,000, Co-PI with Placid Ferreira and Shiv Kapoor.
- **Intel**. Intel Illinois Parallelism Center. \$2,500,000, Co-PI, 2011-2012.
- **Intel**. Leveraging Larrabee's Programmable Rasterization. \$130,000, 2009-2011.
- **Intel/Microsoft**. Universal Parallel Computing Research Center. \$18,000,000, Co-PI, 2008-2012.
- **NAVTEQ**. "Surface Classification and Reconstruction from LIDAR and images." \$60,000, 2008.
- **Thomas M. Siebel**. "MethMorph: Visual Simulation of Methamphetamine Addiction." \$35,000, Spring 2006.
- **NVidia Corp.** \$15,000 annually through the UIUC CS Affiliates Program, 2004–, and \$125,000 total to date in Ph.D. student fellowships (\$25K directly to the student to support one year of Ph.D. research): Nate Carr: 2002 & 2003, Jesse Hall: 2003 & 2004, Jared Hoberock: 2005 & 2008.
- **Microsoft Research**. "Precomputed Radiance Transfer Compression." \$15,000, Sep. 2002.
- **Evans & Sutherland Computer Corp.** "Real Time Procedural Solid Texturing." \$83,246, June 1999 – May 2000.
- **Evans & Sutherland Computer Corp. and Washington Technology Center**. "APST: Computer Graphics Hardware for Antialiased Procedural Solid Texturing." \$77,089. Aug. 16, 1998 - Aug. 15, 1999.
- **Intel Natural Data Types Group**. "Procedural Modeling." \$43,000. Aug. 16, 1997 - Aug. 15, 1998.
- **Intel Natural Data Types Group**. "Recurrent Modeling -- Beyond Fractal Block Coding." (Co-PI with P. Flynn.) \$93,000. Apr. 1, 1995 - Mar. 31, 1997.

From Government Agencies

- **NSF. #OAC-1550554** “SI2-SSI: Collaborative Research: ParaTreet: Parallel Software for Spatial Trees in Simulation & Analysis” (co-PI with Laxmikant Kale), \$170,000, Sep. 2016 – Aug. 2018.
- **NSF. #OCI-1216788** “Collaborative Research: Conceptualizing an Institute for Using Inter-Domain Abstractions to Support Inter-Disciplinary Applications” (co-PI with David Padua and Philippe Geubelle, and other collaborators at Purdue and UT-Austin), \$135,000, Oct. 2012 – Sep. 2016.
- **NSF. #EF-1115112** “Collaborative Research: Digitization TCN: InvertNet--An Integrative Platform for Research on Environmental Change, Species Discovery and Identification” (co-PI with Christopher Dietrich, Christopher Taylor, Nahil Sobh and Umberto Ravaioli), \$2,809,463, July 2011-June 2015.
- **NSF. #OCI-1047764** “SI2-SSE: Collaborative Research: Lagrangian Coherent Structures for Accurate Flow Structure Analysis” (co-PI with Shawn Shadden) \$251,643, Sep. 2010 – Aug. 2013.
- **NSF. #IIS-0534485** “Analysis and Visualization of Complex Graphs” (co-PI with Michael Garland) \$300,000, Sep. 2006 – Aug. 2009.
- **UIUC Critical Research Initiative.** “A New Approach to Bone Replacement.” (Co-PI with Russ Jamison, Michael Goldwasser, Ben Grosser, Matei Stroila and Amy Wagoner Johnson.) \$100,000, Sep. 2005 – Aug. 2006.
- **NSF Small Grant for Exploratory Research.** “Application Directed Surface Parameterization.” \$97,868, Jan. 2005 – Dec. 2005.
- **NSF/DARPA CARGO (Computational and Algorithmic Representations of Geometric Objects) Award.** “Robust Lagrangian Surface Propagation with Topological Control.” (Lead PI, with Michael Heath, Jim Jiao and John Sullivan), \$400,000, May 2003 – May 2006.
- **NSF Information Technology Research. #NSG-0219594** “Making 3D Visibility Practical.” (Co-PI with Steve LaValle, Jeff Erickson, Fredo Durand) \$499,923. Aug. 2002 – Aug. 2005.
- **CNRS (Centre National de la Recherche Scientifique).** Supplement to “Making 3D Visibility Practical” to support UIUC – INRIA collaboration, \$7,000, 2003-5.
- **NSF Information Technology Research. #ACI-0113968** “Multipass Programming for Personal High-Performance Computing.” \$489,671, Aug. 2001 – July 2006.
- **NSF Information Technology Research. #ACI-0121288** “Procedural Representation and Visualization Enabling Personalized Computational Fluid Dynamics.” (Co-PI with David Ebert, David Marcum, Kelly Gaither and Penny Rheingans) \$3,989,773. Aug. 2001 – July 2006.
- **NSF. #NSG-9732379** “Applications of Morse Theory and Catastrophe Theory to Computer Graphics.” (Co-PI with R. Lewis.) \$220,541. Aug. 16, 1997 – Aug. 15 2000.
- **NSF. #CCR-9529809** “Recurrent Modeling.” (Co-PI with P. Flynn.) \$206,435. June 15, 1996 - May 31, 1999. Research Experience for Undergraduates Supplement: \$5,000. Jan. 1, 1998 - May. 31, 1999.
- **NSF Research Initiation Award. #CCR-9309210** “Modeling, Rendering and Animation of Implicit Surfaces.” \$97,925. July 1, 1993 - June 30, 1996. Research Experience for Undergraduates Supplement: \$4,885. Jan. 1, 1995 - Dec. 31, 1995.

Education Projects

- **Nokia University Cooperation Funding.** “Teaching Mobile Augmented Reality on the Windows Phone Platform.” \$11,377.64. 2013.

- **UIUC College of Engineering Strategic Instructional Initiatives Program.** Improvement of key ME courses (Statics, Dynamics and Solid Mechanics), \$450,000, with MechSE Profs. Tortorelli, Dullerud Keane and West, 2012-2014.
- **NSF Special Project.** #EIA-9911033 “The Story of Computer Graphics Documentary Project.” \$48,000. Sept. 15, 1999 – Aug. 31, 2000.

Equipment Grants and Donations

- **Nokia.** Two Velodyne LIDAR scanners, 2013.
- **NVidia.** K20 graphics card, 2012.
- **Intel.** Dell XPS 12, \$1,000, 2012.
- **Intel.** Two Knights Ferry development workstations, \$4,940, 2009.
- **NVidia.** Various graphics cards, 2002-4.
- **ATI.** One graphics card, 2002.
- **Tektronix Phaser 340 Color Printer.** Tektronix. \$5,000. Oct. 1995.
- **High-Performance Networking and Computing Infrastructure for Imaging Research.** (Co-PI with T. Fischer, P. Flynn and R. Bamberger.) NSF Research Instrumentation and Infrastructure. CISE Instrumentation Program, Office of Cross-Disciplinary Activities. \$100,000. March 1, 1994 - April 30, 1996. Awarded: Jan. 1995.
- **Silicon Graphics Inc. Workstation Upgrade.** \$9,000. Apr. 1993.
- **XAOS Tools.** Software Donation. Pandemonium and n-title. \$6,500. Feb. 1993.
- **Karen Guzak.** Art Donation. 12 Prints. \$4,800. May 1995.

Publications

Most of the following publications are available online via
<http://graphics.cs.illinois.edu/~jch/papers>.

Reviewed Publications

1. A.I. Ellis, W. Hunt and J.C. Hart. Real-Time Analytic Antialiased Text for 3-D Environments. Proc. High-Performance Graphics, Computer Graphics Forum 38(8), 2019, pp. 23-32.
2. A.J. Christensen, V. Srinivasan, J.C. Hart and A. Marshall-Colon. Use of computational modeling combined with advanced visualization to develop strategies for the design of crop ideotypes to address food security. Nutrition Reviews 76(5), May 2018, pp. 332-347.
3. P-C Wang, A.I. Ellis, J.C. Hart, C-H Hsu. Optimizing next-generation cloud gaming platforms with planar map streaming and distributed rendering. Proc. NetGames (15th Annual Workshop on Network and Systems Support for Games), 2017.
4. A. Marshall-Colon, S.P. Long, D.K. Allen, G. Allen, D.A. Beard, B. Benes, S. von Caemmerer, A.J. Christensen, D.J. Cox, J.C. Hart, P.M. Hirst, K. Kannan, D.S. Katz, J.P. Lynch, A.J. Millar, B. Panneerselvam, N.D. Price, P. Prusinkiewicz, D. Raila, R.G. Shekar, S. Shrivastava, D. Shukla, V. Srinivasan, M. Stitt, M. Turk, E.O. Voit, Y. Wang, X. Yin, X. Zhu. Crops in silico: Generating Virtual Crops Using an Integrative and Multi-scale Modeling Platform. Frontiers in Plant Science, May 2017.

5. M. Chen, J.C. Hart and S. Shadden. Hierarchical Watershed Ridges for Visualizing Lagrangian Coherent Structures. Reviewed, accepted and presented at TopoInVis: Topology-Based Methods in Visualization, 2015. Published in: Topological Methods in Data Analysis and Visualization IV, H. Carr, C. Garth, T. Weinkauff, Eds. Springer Mathematics and Visualization Series, 2017, pp. 237-252.
6. A.I. Ellis, Warren Hunt and J.C. Hart. "SVGPU: Real Time 3D Rendering to Vector Graphics Formats." Proc. High Performance Graphics, 2016.
7. M. Chen, S. Shadden and J.C. Hart. "Fast Coherent Particle Advection through Time-Varying Unstructured Flow Datasets." IEEE Transactions on Visualization and Computer Graphics 22(8), pp. 1959-1972, 2016.
8. J. Kratt, M. Spicker, A. Guayaquil, M. Fiser, S. Pirk, O. Deussen, J.C. Hart, B. Benes. Woodification: User-Controlled Cambial Growth Modeling. Proc. Eurographics, Computer Graphics Forum 23(2), May 2015, pp. 361-372.
9. V. Lu, J.C. Hart. Multicore Construction of k-d Trees for High Dimensional Point Data. Proc. Advances in Big Data Analytics, July 2014.
10. P.R. Khorrami, V.V. Le, J.C. Hart, T.S. Huang. A System for Monitoring the Engagement of Remote Online Students using Eye Gaze Estimation. Proc. IEEE ICME Workshop on Emerging Multimedia Systems and Applications, July 2014.
11. V. Lu, I. Endres, M. Stroila and J.C. Hart. Accelerating Arrays of Linear Classifiers Using Approximate Range Queries. Proc. Winter Conference on Applications of Computer Vision, Mar. 2014.
12. D. Mayerich, J.C. Hart, Volume Visualization of Serial Electron Microscopy Images Using Local Variance. Proc. BioVis (IEEE Symposium on Biological Data Visualization), pp. 9-16, Oct. 2012
13. C. Dietrich, J. Hart, D. Raila, U. Ravaioli, N. Sobh, O. Sobh, C. Taylor. InvertNet: a new paradigm for digital access to invertebrate collections. ZooKeys 209, July 2012, pp. 165-181.
14. Y. Jia, M. Garland, and J.C. Hart. Social Network Clustering and Visualization using Hierarchical Edge Bundles. Computer Graphics Forum 30(8), Dec. 2011, pp. 2314-2327.
15. Y. Jia, V. Lu, J. Hoberock, M.J. Garland, J.C. Hart. Edge v. Node Parallelism for Graph Centrality Metrics. GPU Computing Gems – Jade Edition, Oct. 2011, pp 15-28.
16. M. Kamali, F.N. Iandola, H. Fang, J.C. Hart. MethMorph: Simulating Facial Deformation due to Methamphetamine Usage. Proc. International Symposium on Visual Computing. Sep. 2011.
17. M. Kamali, J. Cho, M. Stroila, E. Shaffer, J.C. Hart. Robust Classification of Curvilinear and Surface-like Structures in 3D Point Cloud Data. Proc. International Symposium on Visual Computing. Sep. 2011.
18. M. Kamali, E. Ofek, F. Iandola, I. Omer, J.C. Hart. Linear Clutter Removal from Urban Panoramas. Proc. International Symposium on Visual Computing. Sep. 2011.
19. K. Karsch, J.C. Hart. Snaxels on a Plane. Proc. Non-Photorealistic and Artistic Rendering, Aug. 2011.
20. J. Hoberock, J.C. Hart. Arbitrary Importance Functions for Metropolis Light Transport. Computer Graphics Forum 29(6), 2010, pp. 1993-2003.
21. S. Shi, M. Kamali, K. Nahrstedt, J.C. Hart, R. Campbell. High-Quality Zero-Delay Remote Rendering System for 3D Video. Proc. Multimedia, Oct. 2010.
22. B. Choi, R. Komuravelli, V. Lu, H. Sung, R.L. Bochino, S.V. Adve, J.C. Hart. Parallel SAH k-D Tree Construction. Proc. High Performance Graphics, June. 2010.
23. Y. Jia, X. Ni, E. Lorimer, M. Mullan, R. Whitaker, J.C. Hart. RBF Dipole Surface Evolution. Proc. Shape Modeling International, June 2010.
24. W.-W. Feng, B.-U. Kim, Y. Yu, L. Peng, J.C. Hart, Feature-Preserving Triangular Geometry Images for Level-of-Detail Representation of Static and Skinned Meshes, ACM Transactions on Graphics 29(2), March 2010, Article 11.

25. J. Hoberock, V. Lu, Y. Jia, J.C. Hart. Stream Compaction for Deferred Shading. Proc. High Performance Graphics, Aug. 2009.
26. J.C Hart. Assistive Technology for the Aesthetically Impaired. Proc. CHI Workshop on Computational Creativity Support, Apr. 2009.
27. J. Hoberock, S. Hornus, J.C. Hart. On Constructing and Visualizing the Topological Structure of the Visibility and Radiance of Architectural Models. Proc. TopoInVis: Topological Methods in Data Analysis and Visualization, Feb. 2009.
28. Y. Jia, J. Hoberock, M. Garland, J.C. Hart. On the Visualization of Social and other Scale-Free Networks. (Proc. Infovis), IEEE Transactions on Visualization and Computer Graphics 14(6), Nov. 2008, pp. 1285-1292.
29. A. Godiyal, J. Hoberock, M. Garland, J.C. Hart. Rapid Multipole Graph Drawing on the GPU. Proc. Graph Drawing, Sep. 2008.
30. E. Eisemann, H. Winnemoller, J.C. Hart, D. Salesin. Stylized Vector Art from 3D Models with Region Support. (Proc. Eurographics Rendering Symposium), Computer Graphics Forum 27(4), June 2008, pp. 1199-1207.
31. M. Stroila, E. Eisemann, J.C. Hart. Clip Art Rendering of Smooth Isosurfaces. IEEE Transactions on Visualization and Computer Graphics 14(1), Jan. 2008, pp. 71-81.
32. H. Fang, J.C. Hart. Detail Preserving Shape Deformation in Image Editing. Proc. SIGGRAPH, ACM Transactions on Graphics 26(3), Aug. 2007, #12.
33. T. Bergstrom, K. Karahalios, J.C. Hart. Isochords: Visualizing Structure in Music. Proc. Graphics Interface, May 2007.
34. H. Fang, J.C. Hart. RotoTexture: Automated Tools for Texturing Raw Video. IEEE Transactions on Visualization and Computer Graphics 12(6), Nov. 2006, pp. 1580-1589.
35. X. Jiao, A. Colombi, X. Ni, J.C. Hart. Anisotropic Mesh Adaptation for Evolving Triangulated Surfaces. Proc. Meshing Roundtable, Sept. 2006, pp. 173-190.
36. S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, J.C. Hart. Spectral Surface Quadrangulation. Proc. SIGGRAPH, ACM Transactions on Graphics 25(3), July 2006, pp. 1057-1066.
37. N.A. Carr, J. Hoberock, K. Crane, J.C. Hart. Rectangular Multi-Chart Geometry Images. Proc. Sym. on Geom. Proc., July 2006, pp. 181-190.
38. N.A. Carr, J. Hoberock, K. Crane, J.C. Hart. Fast GPU Ray Tracing of Dynamic Meshes using Geometry Images. Proc. Graphics Interface, May 2006, pp. 203-209.
39. P.-T. Bremer, J.C. Hart. A Sampling Theorem for MLS Surfaces. Proc. Point Based Graphics, June 2005.
40. J.O. Talton, N.A. Carr, J.C. Hart. Voronoi Rasterization of Sparse Point Sets. Proc. Point Based Graphics, June 2005.
41. W. Su, J.C. Hart, A Programmable Particle System Framework for Shape Modeling, Proc. Shape Modeling International, June 2005.
42. S. Zelinka, H. Fang, M. Garland, J.C. Hart. Interactive Material Replacement in Photographs. Proc. Graphics Interface, May 2005.
43. S. Hornus, J. Hoberock, S. Lefebvre and J.C. Hart. ZP+: Correct Z-pass Stencil Shadows. Proc. ACM Symposium on Interactive 3-D Graphics and Games, Apr. 2005. **(Highest ranked paper according to review score.)**
44. H. Fang, J.C. Hart, "Textureshop: Texture Synthesis as a Photograph Editing Tool," ACM Transactions on Graphics 23(3) (also in Proc. SIGGRAPH 2004), Aug. 2004, pp. 354-359.
45. X. Ni, M. Garland, J.C. Hart, "Fair Morse Functions for Extracting the Topological Structure of a Surface Mesh," ACM Transactions on Graphics 23(3) (also in Proc. SIGGRAPH 2004), Aug. 2004, pp. 613-622. **(Second highest ranked paper according to tertiary review score.)**

46. N. Carr, J.C. Hart, "Painting Detail," ACM Transactions on Graphics 23(3) (also in Proc. SIGGRAPH 2004), Aug. 2004, pp. 845-852.
47. N. Carr, J.C. Hart, "Dynamic Clustering of Meshed Surfaces," Proc. Symposium on Geometry Processing, July 2004.
48. O.S. Lawlor, J.C. Hart, "Bounding Recursive Procedural Models using Convex Optimization," Proceedings of Pacific Graphics 2003, October 2003, pp. 283-292.
49. P-P. Sloan, J. Hall, J. Hart, J. Snyder. Clustered Principal Components for Precomputed Radiance Transfer. (Proc. SIGGRAPH 2003) ACM Transactions on Graphics 22(3), July 2003.
50. N.A. Carr, J.D. Hall and J.C. Hart, GPU Algorithms for Radiosity and Subsurface Scattering, Proc. Graphics Hardware, July 2003.
51. John C. Hart, Jeyprakash Michaelraj, Brent Baker. Structural Simulation of Tree Growth. The Visual Computer 19(2-3), May 2003, pp. 151-163.
52. P. Sherman, J.C. Hart. Direct Manipulation of Recurrent Models. Computers & Graphics 27(1), Feb. 2003, pp. 143-151.
53. A. Sheffer, J.C. Hart. Seamster: Inconspicuous Low-Distortion Texture Seam Layout. Proc. Visualization 2002, Oct. 2002.
54. N.A. Carr, J.D. Hall and J.C. Hart. The Ray Engine. Proc. Graphics Hardware 2002, Sep. 2002.
55. H. Fang, J. Hart. Randomly Accessible Procedural Animation of Physically Approximate Turbulent Motion. Proc. Computer Animation 2002, June 2002.
56. J. Hart, E. Bachta, W. Jarosz, T. Fleury. Using Particles to Sample and Control More Complex Implicit Surfaces. Proc. Shape Modeling International 2002, May 2002, pp. 129-136.
57. N. Carr, J. Hart. Meshed Atlases for Real-Time Procedural Solid Texturing. ACM Transactions on Graphics 21(2), Apr. 2002.
58. J.C. Hart. Perlin Noise Pixel Shaders. *Proc. Graphics Hardware 2001*, Aug. 2001.
59. W. Cochran, R. Lewis, J. Hart. The Normal of a Fractal Surface. *The Visual Computer* 17(4), 2001, pp. 209-218.
60. J.C. Hart. Using the CW-Complex to Represent the Topological Structure of Implicit Surfaces and Solids. *Proc. Implicit Surfaces '99*, Sept. 1999, pp. 107-112.
61. J.C. Hart, N. Carr, M. Kameya, S.A. Tibbitts, T.J. Coleman. Antialiased Parameterized Solid Texturing Simplified for Consumer-Level Hardware Implementation. *Proc. Graphics Hardware 99*, Aug. 1999, pp. 45-53.
62. J.C. Hart, A. Durr and D. Harsch. Critical points of polynomial metaballs. *Proc. Implicit Surfaces '98*, June 1998, pp. 69-76.
63. J.C. Hart. Morse Theory for Implicit Surface Modeling. In *Mathematical Visualization*, H-C Hege and K. Polthier, Eds., Springer-Verlag, Berlin, Oct. 1998, pp. 257-268. (Reviewed paper presented at VisMath '97, Berlin, 1997.)
64. W.O. Cochran, J.C. Hart, P.J. Flynn. On Approximating Rough Curves with Fractal Functions. *Proc. Graphics Interface '98*, June 1998, pp. 65-72.
65. A.K. Dunker, E. Garner, S. Guillot, P. Romero, K. Albrecht, J. Hart, Z. Obradovic, C. Kissinger and J.E. Villafranca. Protein Disorder and the Evolution of Molecular Recognition: Theory, Predictions and Observations. *Pacific Symposium on Biocomputing* 3: 435-446, Jan. 1998.
66. B. Stander, J.C. Hart. Guaranteeing the Topology of an Implicit Surface Polygonization. *Computer Graphics Annual Conference Series* (Proc. SIGGRAPH 97) Aug., 1997, pp. 279-286.
67. J.C. Hart. Implicit formulations of rough surfaces. *Computer Graphics Forum* 16(2), June 1997, pp. 91-99.
68. B. Burch, J.C. Hart. Linear Fractal Shape Interpolation. *Proc. Graphics Interface '97*. May, 1997, pp. 155-162.

69. J.C. Hart, P.J. Flynn, W.O. Cochran. Similarity Hashing: A model-based vision solution to the inverse problem of recurrent iterated function systems. *Fractals* 5, Apr. 1997, pp. 39-50.
70. W.O. Cochran, J.C. Hart, P.J. Flynn. Fractal volume compression. *IEEE Transactions on Visualization and Computer Graphics* 2(4) Dec. 1996, pp. 313-322.
71. J.C. Hart. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer* 12(10), Dec. 1996, pp. 527-545.
72. S. Worley, J.C. Hart. Hyper-rendering of hyper-textured surfaces. Proc. of *Implicit Surfaces '96*, Oct. 1996, pp. 99-104.
73. J.C. Hart, B. Baker. Implicit Modeling of Tree Surfaces. Proc. of *Implicit Surfaces '96*, Oct. 1996. pp. 143-152.
74. J.C. Hart. Fractal image compression and the inverse problem of recurrent iterated function systems. *IEEE Computer Graphics and Applications* 16(4) July 1996, pp. 25-33.
75. K. Albrecht, J. Hart, A. Shaw, and A.K. Dunker. Quaternion contact ribbons: A new tool for visualizing intra- and intermolecular interactions in proteins. *Pacific Symposium on Biocomputing I*: 41-52, Jan. 1996.
76. D. Hepting, J.C. Hart. The escape buffer: Efficient computation of escape time for linear fractals. Proc. of *Graphics Interface '95*, May 1995. pp. 204-214.
77. J.C. Hart. Implicit formulations of rough surfaces. Proc. *Implicit Surfaces '95*, Apr. 1995, pp. 33-44.
78. B.T. Stander, J.C. Hart. A Lipschitz method for accelerated volume rendering. Proc. of the *1994 Symposium on Volume Visualization*, Oct. 1994. pp. 107-114.
79. J.C. Hart. Distance to an Ellipsoid. Chapter in: *Graphics Gems IV*, P. Heckbert, Ed., Academic Press, 1994, pp. 113-119.
80. J.C. Hart, G.K. Francis, L.H. Kauffman. Visualizing quaternion rotation. *ACM Transactions on Graphics* 13(3) July 1994, pp. 256-276.
81. J.C. Hart. On recording virtual environments. Proc. of *IEEE Visualization '93 Symposium on Research Frontiers in Virtual Reality*, Nov. 1993, pp. 80-83.
82. J.C. Hart, G.W. Lescinsky, D.J. Sandin, T.A. DeFanti, L.H. Kauffman. Scientific and artistic investigation of multidimensional fractals on the AT&T Pixel Machine. *Visual Computer* 7(9) July 1993, pp. 346-355.
83. C. Cruz-Niera, D.J. Sandin, T.A. DeFanti, R.V. Kenyon, J.C. Hart. The CAVE audio visual experience automatic virtual environment. *Communications of the ACM* 35(6) June 1992, pp. 64-72.
84. J.C. Hart. The object instancing paradigm for linear fractal modeling. Proc. of *Graphics Interface '92*, Morgan Kaufman, May 1992, pp. 224-231.
85. J.C. Hart, T.A. DeFanti. Efficient antialiased rendering of 3-D linear fractals. *Computer Graphics* 25(3) (Proc. SIGGRAPH 91) Aug. 1991, pp. 91-100.
86. J.C. Hart, L.H. Kauffman, D.J. Sandin. Interactive visualization of quaternion Julia sets. Proc. of *Visualization '90*. IEEE Computer Society Press. Oct. 1990, pp. 209-218.
87. J.C. Hart, A. Norton. Use of curves in rendering fractures. Proc. of *SPIE/SPSE Symposium on Curves and Surfaces in Computer Vision and Graphics*. Feb. 1990.
88. J.C. Hart, D.J. Sandin, L.H. Kauffman. Ray tracing deterministic 3-D fractals. *Computer Graphics* 23(3) (Proc. SIGGRAPH 89) Aug. 1989, pp. 289-296.

Books, Invited Book Chapters and Edited Course Notes

89. J. Torrellas, S.V. Adve, V.S. Adve, D. Dig, M.N. Do, M.J. Garzaran, J.C. Hart, T.S. Huang, W.W. Hwu, S.T. King, D. Marinov, K. Nahrstedt, D.A. Padua, M. Parthasarathy, S.J. Patel, M. Snir. "Making Parallel Programming Easy: Research Contributions from Illinois," Sep. 2013.

90. M. Olano, W. Heidrich, J. Hart and M. McCool, Real Time Shading, AK Peters, July 2002.
91. J.C. Hart, "Procedural Geometry," Chapter in Modeling and Texturing, a Procedural Approach, 3rd Edition, D. Ebert, Editor, Morgan Kauffman, November 2002.
92. J.C. Hart, "Real-Time Procedural Solid Texturing," Chapter in Modeling and Texturing, a Procedural Approach, 3rd Edition, D. Ebert, Editor, Morgan Kauffman, November 2002.
93. D. Ebert and J. Hart, Editors, "Procedural Implicit Techniques for Modeling and Texturing," ACM SIGGRAPH 1998 Course Notes, July 1998.
94. D. Ebert and J. Hart, Editors, "New Frontiers in Modeling and Texturing," ACM SIGGRAPH 1997 Course Notes, August 1997.
95. J. Hart and D. Saupe, Editors, "Fractal Models for Image Synthesis, Compression and Analysis," ACM SIGGRAPH 1996 Course Notes, August 1996.
96. J. Hart, Editor, "New Directions for Fractal Modeling in Computer Graphics," ACM SIGGRAPH 1994 Course Notes, July, 1994.
97. J. Hart and K. Musgrave, Editors, "Fractal Modeling in 3-D Computer Graphics and Imaging," ACM SIGGRAPH 1991 Course Notes, July 1991.

Patents

98. H. Fang and J.C. Hart, Methods and Systems for Image Modification. #7,365,744. Filed: 26 July 2004, Granted: 29 April 2008.
99. S. Tibbitts, T. Coleman and J.C. Hart. Antialiased Procedural Solid Texturing Hardware. Provisional Patent #60,088,879. Evans & Sutherland, 1998.

Invited Papers, Articles, Tech Reports and Other Publications

100. S.V. Adve, V.S. Adve, G. Agha, M.I. Frank, M.J. Garzaran, J.C. Hart, W.W. Hwu, R.E. Johnson, L.V. Kale, R. Kumar, D. Marinov, K. Nahrstedt, D. Padua, M. Parthasarathy, S.J. Patel, G. Rosu, D. Roth, M. Snir, J. Torrellas, C. Zilles. Parallel Computing Research at Illinois: The UPCRC Agenda. Whitepaper, Nov. 2008.
101. S. Dong, P.-T. Bremer, M. Garland, V. Pascucci and J.C. Hart. Quadrangulating a Mesh using Laplacian Eigenvectors. Tech Rep. UIUCDCS-R-2005-2583, June 2005.
102. P. Lacz and J.C. Hart. Procedural Geometric Synthesis on the GPU. Poster (with accompanying manuscript) for SIGGRAPH 2004 and GP2: The ACM Workshop on General Purpose Computing on Graphics Processors, Aug. 2004.
103. J.D. Hall and J.C. Hart. GPU Acceleration of Iterative Clustering. Poster (with accompanying manuscript) for SIGGRAPH 2004 and GP2: The ACM Workshop on General Purpose Computing on Graphics Processors, Aug. 2004.
104. M. Mullan, R. Whitaker and J.C. Hart. Procedural Level Sets. Manuscript presented at NSF/DARPA CARGO Annual Meeting, May 2004.
105. J.D. Hall, N.A. Carr and J.C. Hart. Cache and Bandwidth Aware Matrix Multiplication on the GPU. Tech. Report UIUCDCS-R-2003-2328, May 2003.
106. M. Kameya, and J.C. Hart. Bresenham noise. SIGGRAPH 2000 Conference Abstracts and Applications, July 2000.
107. J.J. Kim and J.C. Hart. Contour-Based Polygonization of Regular Grid Terrain Data. Proc. Western Computer Graphics Symposium. Mar. 1999.

- 108.A. Aceves, K. Hammil and J.C. Hart. Proposal and Partially Implemented: The Studio: A Large High Resolution Immersive Projection System for Research and Education. Proc. Western Computer Graphics Symposium. Mar. 1999.
- 109.M. Kameya and J.C. Hart. Bit Width Necessary for Solid Texturing Hardware. Proc. Western Computer Graphics Symposium. Mar. 1999.
- 110.N. Carr and J.C. Hart. APST Antialiased Procedural Texturing Interface for OpenGL. Proc. Western Computer Graphics Symposium. Mar. 1999.
- 111.C. Kulratanayan and J.C. Hart. Restructuring the Scene Graph for Efficient Visibility of Natural Scenes. Proc. Western Computer Graphics Symposium. Mar. 1999.
- 112.J.C. Hart. Computational Topology for Shape Modeling. Invited Talk and Paper. Proc. Shape Modeling International '99, Univ. Aizu, Japan, Mar. 1999, pp. 36-45.
- 113.J.C. Hart. Morse Theory for Computer Graphics. WSU Technical Report EECS-97-002. Chapter in ``New Frontiers in Modeling and Texturing," SIGGRAPH '97 course notes, J. Hart, D. Ebert eds., Aug. 1997.
- 114.I.M. Danciu, J.C. Hart. Fractal Color Image Compression in the $L^*a^*b^*$ Uniform Color Space. Poster. Abstract in: Proc. of Data Compression Conference '97. IEEE Computer Society Press, Apr. 1997, p. 434.
- 115.B.T. Stander, J.C. Hart. Topologically Guaranteed Blobby Surface Polygonization. Proceedings of Western Computer Graphics Symposium. Mar. 1996. pp. 1-9.
- 116.J.C. Hart. On the Hyperbolic Plane and Chinese Checkers. Proceedings of Western Computer Graphics Symposium. Mar. 1996. pp. 69-75
- 117.W.O. Cochran, J.C. Hart, P.J. Flynn. Similarity and Affinity Hashing. Proceedings of Western Computer Graphics Symposium. Mar. 1996. pp. 89-98.
- 118.B.T. Stander, J.C. Hart. Interactive re-polygonization of blobby implicit curves. Proceedings of Western Computer Graphics Symposium. Mar. 1995.
- 119.W.O. Cochran, J.C. Hart, P.J. Flynn. Principal component classification for fractal volume compression. Proceedings of Western Computer Graphics Symposium. Mar. 1995.
- 120.J.C. Hart. Sphere tracing: simple robust antialiased rendering of distance-based implicit surfaces. WSU Technical Report EECS-93-015. Chapter in ``Design, Visualization and Animation of Implicit Surfaces," SIGGRAPH '93 course notes, J. Bloomenthal, B. Wyvill, eds., Aug. 1993.
- 121.J.C. Hart. Ray Tracing Implicit Surfaces. WSU Technical Report EECS-93-014. Chapter in ``Design, Visualization, and Animation of Implicit Surfaces," SIGGRAPH '93 intermediate course notes, J. Bloomenthal and B. Wyvill, eds., Aug. 1993.
- 122.T.A. DeFanti, J.C. Hart, D.J. Sandin, L.H. Kauffman. Visualization of multidimensional fractals. Proceedings of the Argonne National Lab. Seventh International Conference on Advanced Science and Technology. Mar. 1991, pp. 289-298.
- 123.T.A. DeFanti, D.J. Sandin, G.W. Lescinsky, J.C. Hart, M.D. Brown, L.H. Kauffman. Simulacra/Stimulacra::Fractal. Art Futura, Jan. 1991.
- 124.J.C. Hart. Parallel algorithms for visualization of multidimensional fractal objects (extended abstract.) Electronic Imaging East '90. Nov. 1990, pp. 459-462.
- 125.D.J. Sandin, J.C. Hart, L.H. Kauffman. Interactive visualization of complex, stacked and quaternion Julia sets. Proceedings of Ausgraph '90, Sep. 1990. (Invited paper.)
- 126.T.A. DeFanti, D.J. Sandin, J.C. Hart, G.W. Lescinsky. One picture's worth a thousand gigaflops (One animation's worth a thousand teraflops). Pixel 1(3) July 1990, pp. 40-41.
- 127.J.C. Hart, A. Norton. Use of curves in rendering fractures. IBM Technical Report. Aug. 1989.

Film and Animation Credits

128. Executive Producer (with C. Machover). "The Story of Computer Graphics." A SIGGRAPH Production of a feature-length documentary on the human struggle to bring computer graphics from the inspired revelation that computers can be used to draw pictures to the present, where low-cost graphics and the Internet combine to provide virtual environments for people to communicate. Premiered at a special event in the Shrine Auditorium, LA, as part of the SIGGRAPH 99 Computer Animation Festival.
129. Contributor. "Air on the Dirac Strings." Produced by L.H. Kauffman, G.K. Francis, D.J. Sandin. SIGGRAPH '93 Electronic Theater. SIGGRAPH Video Review 93, Aug. 1993.
130. Producer. "SIGGRAPH 93 Technical Session Sampler." SIGGRAPH '93 Electronic Theater. Video compilation, narrated by J. Blinn. SIGGRAPH Video Review 91, Aug. 1993
131. Producer. "Highlights from the SIGGRAPH '92 Technical Session." SIGGRAPH '92 Electronic Theater. Video compilation, narrated by J. Blinn.
132. Producer/Director/Animator. "Fun with Octrees: Graph Topologies on the Recurrent Cube." SIGGRAPH '92 Animation Screening Room. SIGGRAPH Video Review 87, Aug. 1992. Appears in: "The Fractal Universe: Animation," Media Magic, 1993.
133. Producer/Director/Animator. "unNatural Phenomena." SIGGRAPH '91 Electronic Theater. SIGGRAPH Video Review 71, Aug. 1991. Appears in: "The Fractal Universe: Animation," Media Magic, 1993. Excerpts appear in: "Beyond The Minds Eye," Mirimar, and "Prime Time Live," ABC.
134. Producer/Director/Animator (with S. Das.). "Sierpinski Blows His Gasket." SIGGRAPH '90 Animation Screening Room. SIGGRAPH Video Review 61, 1990. First Runner-Up, Truevision Videographics Competition, 1990. Appears in: "The Fractal Universe: Animation," Media Magic, 1993.
135. Producer/Director/Animator. "Dynamics in the Quaternions." SIGGRAPH Video Review 42, 1989. Appears in: "The Fractal Universe: Animation," Media Magic, 1993.

∞ Educational Activities ∞

Postdoctoral Associates

- Moussa Mansour, 2015-6. Visiting from Univ. Sao Paolo.
- Matei Stroila, 2005-6. Research Scientist, Navteq, Nokia, Here.
- Peer-Timo Bremer, 2004-6. Research Scientist, Lawrence Livermore National Labs.
- Nate Carr, 2004-5. Senior Research Scientist, Adobe.

Ph.D. Students

- Apollo Ellis. Apple Internship 2016.
- Mingcheng Chen (2016). Currently at Google.
- Victor Lu (2013). Currently at Nokia.
- Mahsa Kameya (2012). Currently at Microsoft. MSR Internship 2010.
- Yuntao Jia (2010). Currently at Facebook.
- Jared Hoberock (2008). Research Scientist, Nvidia. NVidia Fellowship 2005, 2007. NVidia Internship 2005.

- Xinli Ni (2007). Currently at Google, Microsoft Internship 2005.
- Hui Fang (2006). Currently at Google. Adobe Internship 2005.
- Nate Carr (2004). Senior Research Scientist, Adobe. NVidia Fellowship 2002 and 2003. Intel Internship 1999. NVidia Internship 2002.
- Wayne Cochran (1998). Assistant Professor, School of Engineering and Computer Science, Washington State University, Vancouver.
- Bart Stander (1997), Full Professor, Computer and Information Technology, Dixie State College of Utah.

M.S. Students

- 2023: Sharon Lam. “MWEPI-VIZ: An Interactive Visualization Dashboard for Ensemble Simulations in Computational Epidemiology.”
- 2017: Yiwen Xu.
- 2015: Jared Saul.
- 2013: JohnMark Lau.
- 2011: Rajesh Bhasin.
- 2009: Apeksha Godiyal. Microsoft Visio.
- 2005: Jerry Talton (Ph.D. Stanford, 2012), Tony Kaap (Alias), Matei Stroila (NAVTEQ).
- 2004: Jesse Hall (NVidia, Microsoft Research Internship 2002. NVidia Fellowship 2003 and 2004), Patrick Lacz (RadTIME), Wen Su (NVidia).
- 2002: Hui Fang (continued), Ed Bachta (visualization programmer at IUPUI)
- 2000: Nate Carr (continued), Jay Kim.
- 1999: Chanikarn Kulratanayan.
- 1998: Jeyprakash Michaelraj (Alias).
- 1997: Paul Sherman (Periphery Consulting).
- 1996: Ioana Danciu. (Adobe), Brandon Burch (WSU Instructor), Yingjun Yu (Nokia), Wenbiao Jiang (CFD Research), Kurt Albrecht.
- 1994: Anand Ramagopalrao (Microsoft), Wayne Cochran (continued).
- 1992: Yongqi Yang (Adobe).

Courses Taught

UIUC Courses

- CS 498 Data Visualization (via the Coursera platform for the MCS-DS program)
- ENG 198 Grand Challenges (Fall 2012)
- CS 418 Interactive Graphics
- CS 419 Production Graphics
- CS 519 Scientific Visualization

- CS 497 JCH Advanced Shape Modeling (Fall 2000)
- CS 498 Procedural Modeling and Shading in Computer Graphics (Fall 2002)
- CS 498 Particle System Programming (Spring 2004, voluntary addition to course load)
- CS 498 Shape Modeling (Fall 2004)
- CS 598 JCH Surface Reconstruction (2008)
- CS 491 EH Computational Topology Seminar (co-organized with J. Erickson)
- CS 491 GHY Graphics Seminar (co-organized with M. Garland, Y. Yu)

WSU Courses

- CptS 330 Numerical Computing
- CptS 442 Computer Graphics
- CptS 443 Computer-Human Interaction
- CptS 548 Advanced Computer Graphics

Coursera Courses

- Data Visualization, July-Aug. 2015. Enrollment: 88,399 from 198 different countries, including 1,197 with signature track credit toward the Data Mining certificate.

SIGGRAPH Reviewed One-Day Courses

- 2005: “Modern Techniques for Implicit Modeling” (Terry Yoo, James O’Brien), and “GPU Shading and Rendering” (Marc Olano)
- 2004: “Real Time Shading” (Marc Olano)
- 2003: “Beyond Blobs: Recent Advances in Implicit Surfaces” (Terry Yoo, Greg Turk) and “Real Time Shading” (Marc Olano)
- 2002: “Beyond Blobs: Recent Advances in Implicit Surfaces” (Terry Yoo and Greg Turk, org.) and “Real Time Shading” (Marc Olano).
- 2001: “Real Time Shading” (Marc Olano)
- 2000: “Approaches for Procedural Shading on Graphics Hardware” (Marc Olano)
- 1998: “Procedural Implicit Techniques for Modeling and Texturing” (Co-organizer with D. Ebert)
- 1997: “New Frontiers in Modeling and Texturing” (Co-organizer with D. Ebert.)
- 1996: “Fractal Models for Image Synthesis, Compression and Analysis” (Co-organizer with D. Saupe), “Procedural Modeling, Texturing, and Animation Techniques” (D. Ebert) and “Implicit Surfaces for Geometric Modelling and Computer Graphics (Jai Menon and Brian Wyvill)
- 1995: “Procedural Modeling, Texturing, and Animation Techniques” (D. Ebert).
- 1994: “New Directions for Fractal Modeling in Computer Graphics” (Course organizer).
- 1993: “Design, Visualization and Animation of Implicit Surfaces” (J. Bloomenthal, B. Wyvill).
- 1992: “Fractals: from Folk Art to Hyperreality” (P. Prusinkiewicz).
- 1991: “Fractal Modeling in 3-D Computer Graphics and Imaging” (Co-organizer with K. Musgrave).

Other

- Adler Planerarium, Scientific Visualization by Supercomputer. Chicago, Sept.-Nov. 1988.

Invited Colloquia and Presentations

Note: colloquia presented for the purpose of interviewing have been omitted.

- **Symposium on the Future of Data Analytics of the American & Canadian Societies of Animal Science**, invited talk on “Data Visualization,” July 2018.
- **Workshop High Throughput Phenotyping and Data Analytics**, invited talk on “Data Visualization,” Univ. Maryland, Nov. 2017.
- **31st Discovery Conference on Big Data Dairy Management**, invited talk on “Data Visualization,” Nov. 2016.
- **University of Lorraine**, June 2014.
- **University of Notre Dame**, Center for Research Computing, “High Performance Computer Vision, An Oxymoron?,” May 2012.
- **Graphics Interface Conference**, Keynote: Assistive Technology for the Aesthetically Impaired,” May 2011.
- **University of Iowa Computing Conference**, organized and invited by U. Iowa students, “Assistive Technology for the Aesthetically Impaired,” Feb. 2010.
- **AMS MathViz09**, U. Illinois, “It’s Nice to See This Stuff Actually Used for Something,” Mar. 2009. Host: George Francis.
- **INRIA**, Grenoble, France, “Assistive Technology for the Aesthetically Impaired,” Sep. 2009. Host: Elmar Eisemann.
- **Texas A&M University**. Distinguished Lecture: “Assistive Technology for the Aesthetically Impaired,” Feb. 2008. Host: Scott Schaeffer.
- **NAVTEQ**. “Industry Collaborations,” Nov. 2007, Chicago, IL. Host: Matei Stroila.
- **Adobe-MIT Research Retreat**. “Textureshop Redux,” Oct. 2007, Cambridge, MA. Host: David Salesin and Fredo Durand.
- **University of Konstanz Summerschool**. “Computational Topology of Implicit Surfaces” and “Computational Topology of Meshes,” Sep. 2006, Gaschurn, Austria. Host: Dietmar Saupe.
- **Wolfram Research**. “Mathematical Visualization,” May 2006, Champaign, IL. Host: Roger Germundsson.
- **Carnegie-Mellon University**. “Surface Constrained Particle Systems” and “Global Illumination on the GPU,” Feb. 2005. Host: Doug James.
- **Imagina**. “Global Illumination on the GPU,” Monaco, Feb. 2005.
- **3D Festival**. “Global Illumination on the GPU,” Copenhagen, Denmark, May 2004.
- **INRIA**, Grenoble, France, “Global Illumination on the GPU,” Mar. 2004, Host: Samuel Hornus.
- **University of Utah**. “Global Illumination on the GPU,” Oct. 2003, Host: Chuck Hansen.
- **NVidia U**. Keynote speaker. July 2003, Host: Herbert Hu.

- **Army High Performance Computing Research Center (AHPARC) Workshop on Graphics Modeling, Simulation and Visualization.** “Harnessing the Supercomputer in your PC’s Graphics Card,” June 2003. Host: Boquan Chen.
- **University of Illinois at Chicago (Math Dept.).** “Computational Topology for Computer Graphics.” Apr. 2002. Host: Anatoly Libgaber.
- **Colorado Springs SIGGRAPH Chapter.** “The Quest for Real-Time Procedural Solid Texturing.” Mar. 2001. Host: Dino Schweitzer.
- **Northwestern University.** “The Quest for Real-Time Procedural Solid Texturing.” Oct. 2000. Host: Benjamin Watson.
- **Shape Modeling International.** Invited Speaker, Aizu, Japan, March 1999.
- **Universität Freiburg.** “Recurrent Modeling.” Sep. 1997. Host: Dietmar Saupe.
- **University of Utah.** “The Topology of Implicit Surfaces.” Feb. 1997. Host: Chuck Hansen.
- **University of Illinois at Chicago (EECS Dept.).** “The Topology of Implicit Surfaces.” Feb. 1997. Host: Tom DeFanti.
- **Interval Research.** “Recurrent Modeling.” July 1996. Host: Sally Rosenthal.
- **University of Washington.** “Recurrent Modeling.” Dept. of Computer Science and Engineering. Jan. 1996. Host: Linda Shapiro.
- **Boeing Information & Support Services.** “Why Fractal Image Compression Isn’t Fractal.” Computer Graphics & Multimedia Group. Jan. 1996. Host: David Kerlick.
- **University of Idaho.** “Why Fractal Image Compression Isn’t Fractal.” Computer Science Dept. Nov. 1995. Host: Deborah Frincke.
- **NATO Advanced Study Institute** on Fractal Image Compression and Image Analysis. “Similarity Hashing: A model-based vision solution to the inverse problem of recurrent iterated function systems.” July 1995.
- **Simon Fraser University.** “Recurrent Modeling.” School of Computing Science. July 1995. Host: F. David Fracchia.
- **University of Calgary.** “Procedural Geometric Instancing.” Dept. of Computer Science. Jan. 1994. Host: Przemyslaw Prusinkiewicz.
- **Purdue University.** “Linear Fractals in Computer Graphics.” Computer Science Dept. June 1992. Host: Chandrajit Bajaj.

Professional Activities

Editorships and Papers Chairships

- Area Editor (Visual Computing) and Editorial Board Member, ACM Books 2014-2022.
- Papers Co-Chair (with Scott Schaefer), Shape Modeling International, 2012.
- Editor-in-Chief, ACM Transactions on Graphics, 2002-2008. Past EiC, 2008-2011.
- Associate Editor, ACM Transactions on Graphics, 2000-2002, 2011-.
- Guest Editor, Communications of the ACM 35(6) June 1992.

- Guest Editor, SIGGRAPH Video Review 96, ``Video Supplement to the Conference Proceedings," Aug. 1993.
- Guest Editor (with E.E. Catmull), SIGGRAPH Video Review 86, ``Video Supplement to the Conference Proceedings," July 1992.

Professional Organization Leadership

- SIGGRAPH Small Conferences Committee, 2010-
- Chair, SIGGRAPH Service Award, 2005-8.
- ACM SIGGRAPH Advisory Board, 2005.
- Director for Communications, SIGGRAPH Executive Committee, 1996-9.
- Director at Large, SIGGRAPH Executive Committee, 1994-6.

External Research Center Review Panels

- King Abdullah University of Science and Technology, Mar. 2012.
- Science Foundation Ireland, Oct. 2006.
- INRIA Systemes Cognitifs D, May 2006.

Conferences Chaired

- Co-Chair (with A. Marshall-Colon, S.P. Long and E.H. DeLucia), Plants-in-Silico, NCSA, UIUC, 2016.
- Co-Chair (with S. Gao, J. Jorge, K. Polthier and W. Wang), Shape Modeling International, Hong Kong, Oct. 2014.
- Co-Chair (with Marie-Paule Cani), Shape Modeling International, MIT, June 2005.
- Co-Chair (with K. van Overveld), Implicit Surfaces, Eindhoven, Netherlands, 1996.
- Chair. Western Computer Graphics Symposium, Panorama, BC, 1996.

Program Committees

- SIGGRAPH Papers Committee: 2001, 2006, 2007, 2010
- SIGGRAPH Asia Papers Committee, 2008.
- SIGGRAPH Sketches Jury: 1998, 1999.
- Symposium on Geometry Processing, all since 2006.
- Interactive 3D Graphics and Games, all since 2005.
- Pacific Graphics, all since 2002.
- Shape Modeling International, all since 1997.
- Implicit Surfaces (all).
- Eurographics 2004.

- Graphics Hardware, all since 2004.
- Web3D 2003.
- IEEE Virtual Reality 2000, 2001, 2002.
- Virtual Reality Annual International Symposium 1998.
- Visualization and Mathematics 1997.
- Supercomputing 1991-3, 1995.

Memberships

- Association for Computing Machinery
- ACM Special Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH)
- IEEE Computer Society

University of Illinois Committees

University of Illinois System

- Steering Committee, Start MyResearch Portal for university system-wide research management software, 2014-2016.
- University Technology Management Team, 2012-2013.

Urbana-Champaign Campus

- Ex-Officio (Graduate College Representative), Senate Educational Policy Committee, 2014-.
- Chair, Senate Subcommittee on IT, 2013-6, Member, 2008-2010.
- University Senate Executive Committee, 2013-6.
- IT Governance Executive Committee, 2012-5.
- Chair, IT Governance Research Subcommittee, 2012-5.
- Chair, Stewarding Excellence IT Project Team, 2010.
- Human Factors Transition Committee, 2008-9.
- Senate, 2007-14,

College of Engineering

- Chair, Promotion & Tenure Committee, 2013, Member 2012-2013.
- Vice-Chair, Executive Committee, 2013. Secretary, 2012-2013. Member, 2011-2013.
- Chair, Engineering IT Steering Committee, 2010-11.

Dept. of Computer Science

- Area Chair, Graphics and Human-Computer Interaction, 2004-2011.
- Department Head Search Committee, 2007-9.
- Advisory Committee (peer elected), 2005-2006.
- Industry Relations Committee, 2005-2007.
- Technology, Facilities and Services Committee (CS), 2005-2012.
- Engineering Open House CS Representative (CS), 2001-2006.
- Recruiting Committee, 2004, 2005.
- Curriculum Committee, 2002-3.
- Chair, Distinguished Lecture Series 2001-2002, Member 2000-2002.
- Computer Systems (CS), 2000-2001.
- Fellowships, Assistantships and Admissions, 2000-2001.

WSU

- School of EECS Computer Science Area Chair, 1998-2000.
- School of EECS Computer Science Accreditation Coordinator, 1999-2000.
- School of EECS Curriculum Committee 1998- (Chair, 1999-2000).
- School of EECS Graduate Studies Committee 1993-8.
- School of EECS Laboratory Committee 1994-6.
- WSU Center for Visualization, Analysis and Design in the Molecular Sciences (VADMS) Advisory Board 1994-2000.