

Case No. IPR2023-00688
U.S. Patent No. 11,184,628

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. IPR2023-00688
U.S. Patent No. 11,184,628

**DECLARATION OF ANSELMO LASTRA IN SUPPORT OF
PATENT OWNER'S PRELIMINARY RESPONSE TO
INTER PARTES REVIEW OF U.S. PATENT NO. 11,184,628**

TABLE OF CONTENTS

	PAGE
I. INTRODUCTION	1
II. QUALIFICATIONS AND EXPERIENCE.....	2
III. MATERIALS CONSIDERED	7
IV. LEGAL STANDARDS	9
A. Claim Construction	10
B. Obviousness.....	10
V. PERSON OF ORDINARY SKILL IN THE ART	12
VI. CLAIM CONSTRUCTION	13
VII. RELEVANT TECHNICAL BACKGROUND AT THE TIME OF THE '628 PATENT	13
A. Graphics Processing	14
B. Texture Mapping and Texels.....	15
C. Texture Compression	16
1. Images and Blocks	16
2. Graphics Processing Versus General-Purpose Processing	18
VIII. U.S. PATENT NO. 11,184,628	19
A. Drawbacks in Prior Art Texture Compression Methods.....	20
B. The '628 Patent's Disclosed Solutions	22
1. Disjoint Subsets.....	22
2. Index Values	25
IX. DESCRIPTION OF ASSERTED ART.....	28
A. Sorgard (Ex. 1004)	28
B. FXT1 (Ex. 1005)	32
X. EVALUATION OF PETITIONER'S PROPOSED GROUNDS	33
A. Ground 1: Sorgard and the Knowledge of a POSITA Does Not Render Claims 7-12 of the '628 Patent Obvious	33
1. Sorgard's Fading Pattern Index Is Not Either of the Claimed Index Values.....	33

2.	Dr. Manocha Fails to Identify a Second Index Value	37
B.	Ground 2 and 4: FXT1 and the Knowledge of a POSITA Does Not Render Claims 7-12 Obvious, Nor Does the Addition of Resehtov	39
1.	Construction of “Block” and “Subset”	39
2.	Under the Correct Construction, Dr. Manocha Fails to Allege that FXT1 Discloses “Subsets”	44
3.	Petitioner’s Suggested Modifications Would Render FXT1 Inoperable	45
4.	Dr. Manocha’s Other Proposed Modification to FXT1 is Hindsight Without Any Motivation	47
XI.	CONCLUSION.....	48

I, Anselmo Lastra, 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. 11,184,628 (the “’628 Patent”) (Ex. 1001) in IPR2023-00688.

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 Dinesh Manocha (“Manocha Decl.”) (Ex. 1003), which I understand was submitted in support of Realtek’s Petition for *Inter Partes* Review of the ’628 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 in the Department of Computer Science at the University of North Carolina at Chapel Hill since 1991. I am currently an Emeritus Professor and served as Chairman of the Computer Science Department from 2009 to 2014.

5. I received my Bachelor of Science degree in Electrical Engineering from the Georgia Institute of Technology in 1972. In 1981, I received a Master of Arts degree in Computer Science from Duke University and in 1988, I received a Ph.D. in Computer Science from Duke University.

6. Since 1972, I have been involved in the research, development, and design of hardware and software for computer systems, including computer graphics and image-based rendering systems.

7. From 1972 to 1975, I was an engineer at Scidata Inc. in New York, New York and Miami, Florida. From 1975 to 1979, I was an engineer and then project manager at Coulter Electronics in Miami, Florida. While holding these positions, I designed electronic hardware, software, and medical instruments and for a time supervised a team of engineers.

8. In 1986, I was a consultant at AT&T Bell Laboratories. I worked as an engineer on a project to develop a special purpose dataflow machine for processing sonar signals to automatically identify submarines and surface vessels. From 1988

to 1991, I was a Research Assistant Professor at Duke University in Durham, North Carolina. While holding this position, I taught computer science classes and researched the mathematics of genetics and problems relating to cardiac conduction irregularities that can lead to heart arrhythmias.

9. In 1991, I joined the Computer Science Department at the University of North Carolina at Chapel Hill as a Research Assistant Professor. In 2001, I was promoted to Associate Professor. In 2006, I was promoted to full Professor, and from 2009 to 2014 served as Chairman of the Computer Science Department.

10. At the University of North Carolina at Chapel Hill, I worked on the Pixel-Planes, PixelFlow, and WarpEngine projects. With support from the National Science Foundation and the Defense Advanced Research Projects Agency, we developed a heterogeneous graphics supercomputer called Pixel-Planes 5 that was, to the best of my knowledge, the fastest of its time. Pixel-Planes 5 was benchmarked on the SPEC “head” dataset at a performance four times that of any existing commercial graphics workstation. In collaboration with the Hewlett-Packard Corporation, we later developed another heterogeneous graphics supercomputer called PixelFlow. To the best of my knowledge, PixelFlow was also several times faster than any graphics workstation of its time.

11. Pixel-Planes 5 and PixelFlow were tiled architectures and implemented deferred shading. Both machines were fully programmable, including at the

fragment/pixel level. On Pixel-Planes 5, we programmed the lighting and shading in assembly language, and on PixelFlow, we introduced a high-level language for real-time shading and lighting¹. Our intent was to enable shading like Pixar was doing but in real time instead of offline. To the best of my knowledge, that was the first graphics accelerator to enable fragment shading (the computation of pixel color) in a high-level programming language. It took almost a decade for commercial products to follow suit and now essentially all devices with a display (computers, smart phones, automobile entertainment systems, etc.) support this feature.

12. In 1997, after completing the PixelFlow machine, our team decided to research approaches to perform image-based rendering of synthetic and real-world scenes in real time. This work was inspired by the work of our colleagues McMillan and Bishop². We designed a hardware architecture, which we named the WarpEngine, for real-time image-based rendering³. The notion was to render from

¹ Anselmo Lastra, Steven Molnar, Marc Olano, and Yulan Wang. 1995. Real-time programmable shading. In Proceedings of the 1995 symposium on Interactive 3D graphics (I3D '95). Association for Computing Machinery, New York, NY, USA, 59. DOI: <https://doi.org/10.1145/199404.199414>.

² Leonard McMillan and Gary Bishop. 1995. Plenoptic modeling: an image-based rendering system. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (SIGGRAPH '95). Association for Computing Machinery, New York, NY, USA, 39–46. DOI: <https://doi.org/10.1145/218380.218398>.

³ Voicu Popescu, John Eyles, Anselmo Lastra, Joshua Steinhurst, Nick England, and Lars Nyland. 2000. The WarpEngine: an architecture for the post-polygonal age. In Proceedings of the 27th annual conference on Computer graphics and

arbitrary views, with input from camera images (with depth) of natural environments.

13. In 2002, we pioneered the use of energy-efficient graphics hardware for general purpose computation, a technique that we called General Purpose Computing on Graphics Processors, or GPGPU⁴. This is now the basis for many of the fastest supercomputers in the world, which use many graphics processing units along with standard central processing units (in fact, the GPU business has bifurcated with GPUs for data center servers marketed separately from those for personal computers, phones, etc.). Two colleagues and I, in 2004, chaired a new workshop to bring together researchers from around the world to discuss how to launch GPGPU into common usage.

14. At the University of North Carolina at Chapel Hill, I taught courses in computer graphics, graphics hardware, computer architecture, 3D computer animation, image generation, robotics, digital logic, programming, and other subjects.

interactive techniques (SIGGRAPH '00), 433–442. DOI:
<https://doi.org/10.1145/344779.344979>.

⁴ Mark J. Harris, Greg Coombe, Thorsten Scheuermann, and Anselmo Lastra. 2002. Physically-based visual simulation on graphics hardware. In Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware (HWWS '02). Eurographics Association, Goslar, DEU, 109–118.

15. I am a named inventor on two U.S. Patents. These patents are generally related to computer graphics and image-based rendering.

16. I have authored or co-authored over 80 research papers and articles, many of which relate to computer graphics and image-based rendering, and were presented in prestigious venues, such as the annual conference of the Association for Computing Machinery's Special Interest Group on Computer Graphics and Interactive Techniques ("SIGGRAPH").

17. I served a term on the editorial board of the IEEE Transactions on Visualization and Computer Graphics. I also served a term on the editorial board of IEEE Computer Graphics and Applications. I was on the SIGGRAPH conference organization committee. I chaired the Symposium on Interactive 3D Graphics and served on the advisory committee. I served as papers chair, program chair, treasurer and general chair of the Graphics Hardware conference, and served on the advisory board. In 2009, we merged Graphics Hardware with the Symposium on Interactive Ray Tracing and created the High Performance Graphics Conference. Until this year, I continued to serve on the steering committee of the High Performance Graphics Conference.

18. I am being compensated for my time working on this matter at my usual rate of \$600 per hour. My compensation does not depend on the outcome of this litigation.

19. 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

20. 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- 00688	Description
Ex. 1001	U.S. Patent 11,184,628 to Iourcha et al.
Ex. 1002	Prosecution History of U.S. Patent 11,184,628
Ex. 1003	Declaration of Dinesh Manocha, Ph.D.
Ex. 1004	U.S. Patent No. 8,102,402 (“Sorgard”)
Ex. 1005	FXT1 Texture Compression Technology White Paper (“FXT1”)
Ex. 1006	U.S. Patent No. 6,683,978 (“Iourcha ’978”)
Ex. 1007	<i>Image Compression Using Block Truncation Coding</i> (“Delp”)
Ex. 1008	U.S. Patent Pub. No. 2004/0151372 to Reshetov (“Reshetov”)
Ex. 1009	U.S. Patent No. 7,388,993 (“Govindaswamy”)
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	Chen et al., <i>A JPEG-like texture compression with adaptive quantization for 3D graphics application</i> , the Visual Computer, October 2, 2001

Exhibit No. IPR2023- 00688	Description
Ex. 1012	Stachera et al., <i>Fractal-based hierarchical mip-pyramid texture compression</i> , Machine Graphics and Vision, February 2006
Ex. 1013	https://web.archive.org/web/20041120003604/http://www3.sharkyextreme.com/hardware/articles/99/3dfx_fxt/wp/4.shtml
Ex. 1014	<i>The New Voodoos</i> , PC Gamer, Vol. 7 No. 2, February 2000
Ex. 1015	Gizmopolis, Computer Games Magazine, No. 109, December 1999
Ex. 1016	U.S. Patent Pub. No. 2004/0228527 (“Iourcha ’527”)
Ex. 1017	Levkovich-Maslyuk, <i>Texture compression with adaptive block partitions</i> , ACM Multimedia 2000
Ex. 1018	Hesseldahl, <i>Do You Voodoo Like I Do?</i> , Electronic News Online, November 22, 1999
Ex. 1019	Exhibit 43 to ITC Complaint, Investigation No. 337-TA-1318, Certain Graphics Systems, Components Thereof, and Digital Televisions Containing the Same
Ex. 2002	James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. 1990. <i>Computer Graphics: Principles And Practice</i> (2nd ed.). Addison-Wesley Longman Publishing Co., Inc.
Ex. 2003	Edwin Earl Catmull. 1974. “A Subdivision Algorithm For Computer Display of Curved Surfaces.” Ph.D. Dissertation. The University of Utah
Ex. 2004	Ziyad S. Hakura and Anoop Gupta. 1997. The design and analysis of a cache architecture for texture mapping. SIGARCH Comput. Archit. News 25, 2 (May 1997), 108–120.
Ex. 2005	Michael Doggett. 2012 “Texture Caches” In: IEEE Micro. Vol. 32, No. 3. pp. 136-141.
Ex. 2006	fsdeveloper.com wiki (https://www.fsdeveloper.com/wiki/index.php/Texture_formats_overview)
Ex. 2007	OpenGL Wiki (www.khronos.org/openg/wiki/S3_Texture_Compression)

Exhibit No. IPR2023- 00688	Description
Ex. 2008	Rui Wang Wei Hua, et al. 2012. Compressing repeated content within large-scale remote sensing images. Vis. Comput. 28, 6–8 (June 2012), 755–764.

21. 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

22. In preparing and expressing my opinions and considering the subject matter of the '628 Patent, I am relying on certain basic legal principles that counsel have explained to me. These principles are discussed below.

A. Claim Construction

23. 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).

24. 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

25. 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.

26. 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.

27. 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.

28. 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.

29. 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. PERSON OF ORDINARY SKILL IN THE ART

30. I have been asked for the purposes of this declaration to opine on the knowledge and understanding of one of ordinary skill in the art as of August 31, 2006.

31. I have been informed and understand that a person of ordinary skill in the art (“POSITA”) is a hypothetical person who is presumed to have been familiar with the relevant art at the time of the invention. I have also been informed and understand that, in determining the level of skill in the art, one may consider: the type of problems encountered in the art, prior art solutions to those problems, the rapidity with which innovations are made, the sophistication of the technology, and the educational level of active workers in the field.

32. I understand the ALJ in ITC Investigation No. 337-TA-1318 has adopted the following definition of a person of ordinary skill in the art (“POSITA”): “a POSITA would have a bachelor degree in electrical engineering, computer engineering, computer science, or a related field, and at least two years of experience in ‘parallel computing or graphics processing including developing, designing, or programming software or hardware for graphics processing or parallel processing

units, graphics accelerators[,] or other graphics processing systems,’ with additional education substitution for experience and vice versa.” Ex. 1010 at 9 (alteration in original).

33. I meet these criteria, and any other appropriate criteria of a POSITA for the ’638 Patent, and I consider myself a POSITA pertaining to the ’628 Patent. In addition, I would have been a POSITA at the time of the inventions of the ’628 Patent.

VI. CLAIM CONSTRUCTION

34. I understand that the ALJ has construed the term “unpacked” to mean “extracted.” Ex. 1010 at 45-46. I understand that Dr. Manocha applied this construction in rendering his opinion. Ex. 1003, ¶ 44. I have also applied this construction for the purposes of rendering my opinion.

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

VII. RELEVANT TECHNICAL BACKGROUND AT THE TIME OF THE ’628 PATENT

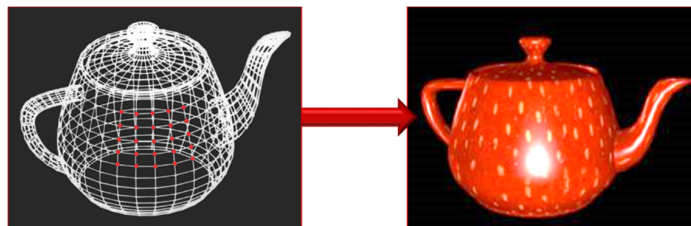
36. To the extent it may be helpful to the Board in understanding the context of the ’628 Patent and in framing my opinions, I offer the following summary of pertinent technical information that would have been available to a person of ordinary skill in the art as of August 31, 2006, which I understand is the earliest

priority date for the '628 Patent and is thus the lens through which the subject matter is to be evaluated.

37. The challenged U.S. Patent No. 11,184,628 relates generally to the field of texture compression and decompression in computer graphics processing. For example, the '628 Patent discloses improved texture compression and decompression techniques where texture blocks are subdivided into disjoint subsets and interpolation is used to determine texel colors.

A. Graphics Processing

38. Graphics processing involves manipulating and rendering 3D graphical objects onto a 2D display screen. Specifically, 3D objects (typically represented by a collection of geometrical shapes such as triangles) are converted into a 2D image (typically comprising a collection of pixels) as shown below. Ex. 2002 at 3, 11-15, 17-20. 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. 2002 at 210-222, 722-741.

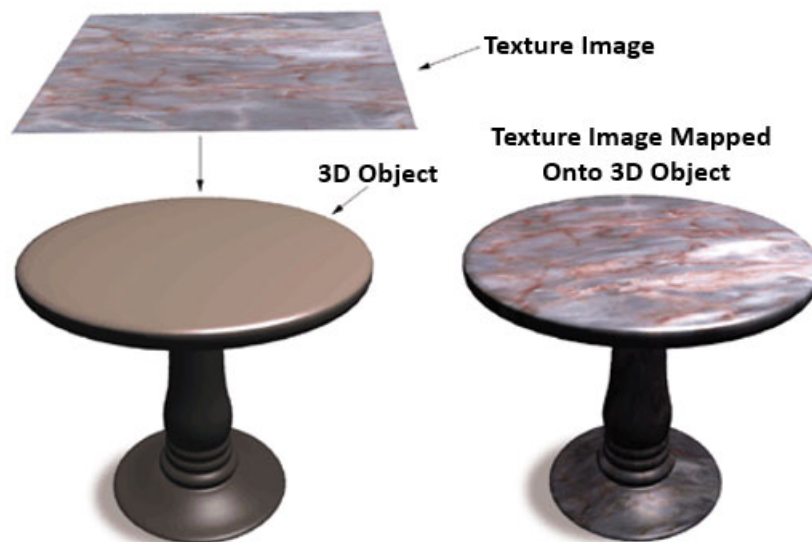


39. The conversion of a 3D shape consisting of a mesh of triangles into a photorealistic 2D image consisting of an array of pixels happens in a number of

processing stages. Ex. 2002 at 806-809, 866-871. The “pixel processing” or “shading” stage relevant here includes the computation of the colors of each of these pixels/fragments, as well as a process called “texture mapping.” Ex. 2002 at 868-872; *see generally* Ex. 2003.

B. Texture Mapping and Texels

40. One aspect of converting 3D objects into 2D images involves texture mapping. *See* Ex. 1001 at 3:61-4:2; Ex. 1004 at 1:13-26. Texture mapping is the process of applying a texture image to a group of pixels representing a surface to be displayed. *Id.* at 3:61-4:2. Texture mapping reproduces the detail by mapping a texture image onto these pixels. *Id.* For example, in the illustration below, a texture image (the marble pattern) is mapped onto a group of pixels representing 3D object (a table) to create the image of a marble table.



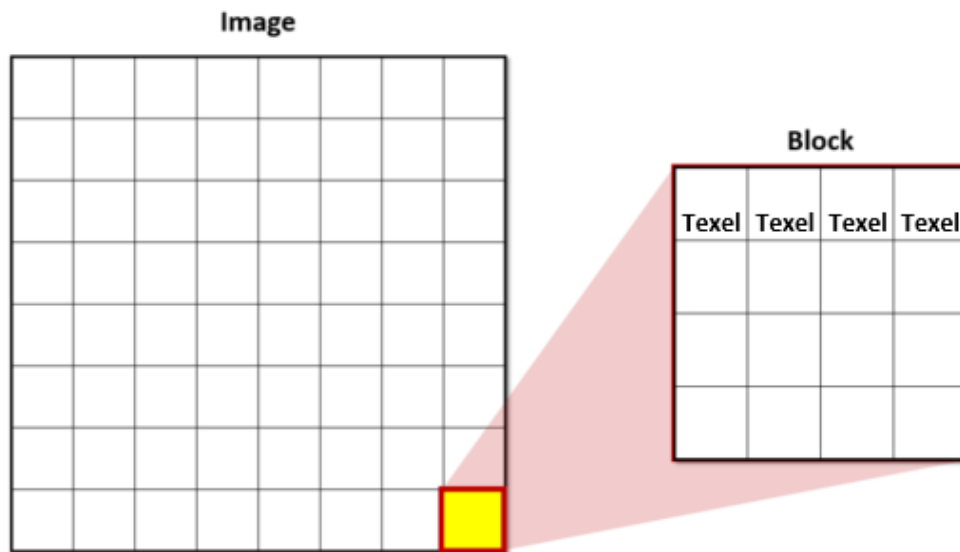
41. The texture image is made up of “texels.” Ex. 1001 at 3:48-53. Texels are similar to pixels: the same way a “pixel” is an element of a picture, a texel is an element of a texture image. *Id.* The texel represents the color or other surface characteristic of a corresponding pixel on the surface to be displayed. *Id.* at 3:65-4:2.

C. Texture Compression

42. Fetching texels from the external memory of a GPU for texture mapping consumes a great amount of bandwidth, so texture blocks are usually stored in a compressed format and then decompressed on the GPU as needed. *See* Ex. 1001 at 4:3-15.

1. Images and Blocks

43. In one prior art approach, for the GPU to be able to load texels as needed, the texture image is divided into fixed size blocks (*e.g.*, 4x4 or 4x8, similar to the below image) that can be individually fetched as needed, rather than fetching the entire texture image. *See generally* Ex. 1001 at 1:41-2:52 (describing prior art methods using blocks), 5:65-6:4; *see also* Ex. 1011 at 2.



44. This block-based compression differs from general image compression, such as JPEG, where the size of the compressed image can be variable because the whole image is meant to be loaded at once. That whole-image approach presents difficulties with random access of blocks, which is a necessity with texture mapping. *See* Ex. 1001 at 2:12-23; Ex. 1011 at 2 (“Although textures are ordinary bitmap images, current image compression methods may not be applicable to [texture compression]. The nature of texture access make [texture compression] differ from traditional image compression.”), 4 (providing that JPEG compression poses a “random access problem” when used with texture compression); Ex. 1012 at 2 (“[M]ost classical image compression methods cannot be applied to textures.”).

2. Graphics Processing Versus General-Purpose Processing

45. Graphics processing, and especially real-time graphics on graphics hardware (GPUs), differs from general-purpose processing for several reasons. One important distinction is memory constraints. *See* Ex. 1001 at 4:3-15; Ex. 1011 at 1. It is impractical or impossible to load a full texture image (which may be large, *e.g.*, 2048 x 2048 texels) into GPU internal memory because of memory size limitations, as Petitioner's own asserted art states. *See id.*; *see* Ex. 1005 at 2 ("One of the issues when using large numbers of textures is simply that it takes much more memory to store those textures. Consider, for example, that a 2048x2048 32-bit per texel texture requires 16 Mbytes of texture storage space Furthermore, this is only for a single high resolution texture. It wasn't long ago that 3D accelerators did not even have 16 Mbytes of total memory onboard."). That number is for external memory, which is much larger than internal memory.

46. This was especially true in 2006, the year of priority of the '628 patent, when the number of transistors in a GPU was much more limited than it is today. Multiple textures are normally being used at once, further constraining internal GPU memory. *See* Ex. 1001 at 1:25-28; 4:3-15. As a result, the block sizes need to be small so that one or more portions of textures can be accessed without consuming too much memory, and certainly operating on the entire image at once is not possible. This is true today, and was especially true in 2006.

47. Internal texture memory in a GPU is organized as a cache, where older blocks are discarded to make room for more recently accessed texture blocks, and it is advantageous to have each cache line be the size of a compressed texture block. Ex. 2004 at 114. The caches that store the compressed texture blocks are small. Even as late as 2012, Doggett reported that level 1 caches were between **12 and 16 kilobytes**. Ex. 2005 at 139. This means that in 2006, level 1 caches would have been even smaller. Level 2 caches are somewhat larger, but are shared with other data, such as geometry, final pixels, etc. As an example of what could be held in a cache memory on a GPU, we can look at FXT1, where a single compressed 256x256 image (in a 4 bit-per-texel format) would consume 32KB of space, which is approximately twice the size of a typical level 1 cache in 2012. Ex. 1005 at 2.

48. Furthermore, texture mapping often uses only parts of the entire texture image, depending on which positions of the scene are to be rendered. *See* Ex. 1011 at 2 (discussing the need for random access of texture data); Ex. 1012 at 3 (same). Decompressing a full texture image when only a small component is needed would be wasteful and is not done. *See id.*

VIII. U.S. PATENT NO. 11,184,628

49. The utility application that gave rise to the '628 Patent, Application No. 16,257,500, was filed on January 25, 2019, and claims the benefit of original utility application 11/513,190, filed on August 31, 2006, through a series of continuation

applications. I understand that the '628 Patent is entitled to a priority date no later than the filing of the utility application on August 31, 2006.

50. The '628 Patent discloses improved texture compression and decompression techniques where texture blocks are subdivided into disjoint subsets and individual index values are used to interpolate and determine the texel colors of the respective subsets.

A. Drawbacks in Prior Art Texture Compression Methods

51. The '628 Patent discusses several different prior approaches to texture compression and identifies some of the drawbacks of each. Ex. 1001 at 1:31-40. Such drawbacks include visible artifacts, gradient noise, blurry images, loss of high frequency details, incompatibility with fixed rate compression schemes, and issues with textures comprising numerous colors. *See generally id.* at 1:41-2:65.

52. For example, among the prior art approaches discussed and distinguished are other block-based techniques such as DXTC and FXT1. *Id.* at 2:24-52. As the '628 Patent discloses, DXTC “is a block-based texture compression scheme” that divides a texture image into 4x4 pixel blocks. *Id.* at 2:24-32. DXTC does not have more than one level of division – that is, it does not further divide the blocks into subsets. DXTC is also known as DXTn (where n represents a number 1 through 5 and indicates different variations of the algorithm) and was originally known as S3TC.

53. FXT1 is another block-based texture compression scheme that “essentially extends DXTC with some additional block types that can be mixed within any given image, and also provides a 4 bits per pixel compression mode for textures with alpha.” *Id.* at 2:46-50. The additional block types use slightly different compression algorithms, adding, for example, variable transparency for some block types and no interpolation of color for others. As with DXTC, there is a single level of division from the entire image to blocks in FXT1, meaning that FXT1’s additions to DXTC did not add an additional level of division. *See* Ex. 1005 at 5.

54. The DXTC and FXT1 block-based approaches suffered from numerous drawbacks. For example, DXTC has difficulty with textures that have many different color hues within one block and textures containing blocks that have multiple separate color gradients at different orientations. Ex. 1001 at 2:34-44. This is because the color palette in DXTC is limited, for example, to a single interpolated line in color space. Therefore, the colors are related. As a result, on blocks with multiple color gradients, DXTC sacrifices accurate compression of one of the gradients. *Id.* Additionally, DXTC’s low precision of the color endpoints and small number of interpolants produces significant noise for some gradients, particularly ones that are oriented diagonally to the pixel blocks. *Id.*

55. The FXT1 block-based scheme did not provide any image quality gains over DXTC. As a result, it received limited industry support. *Id.* at 2:46-52.

B. The '628 Patent's Disclosed Solutions

56. The '628 Patent significantly advanced block-based texture compression and decompression and addressed the foregoing drawbacks in numerous ways, including: (i) by dividing each texture image into blocks, and further dividing each block into disjoint subsets; and (ii) using individual indexes for each texel of a subset to assign and determine the color for the respective texel. *Id.* at 6:5-10, 10:61-11:11.

1. Disjoint Subsets

57. The '628 Patent describes that while prior art methods, such as DXTC and FXT1, compressed images into blocks of pixels (*e.g.*, 4x4 or 8x8 blocks), the '628 Patent discloses inventive embodiments that further subdivide the blocks into “subsets” of varying shapes and sizes. *Id.* at 6:5-15, 27-42.

58. Specifically, the '628 Patent describes inventive embodiments as a refinement of the DXTC and FXT1 type of block-based compression, one that further partitions texture blocks:

To address this challenge, the present invention further refines block-based image compression techniques such as DXTC with a flexible method for pre-compression partitioning of texture blocks.

Id. at 4:41-58; 5:66-6:1.

59. As the '628 Patent further describes, rather than merely dividing a texture image into smaller rectangular blocks like the DXTC and FXT1-type approaches, this inventive embodiment further divides those blocks into subsets:

As noted above, the present invention represents a further refinement of block-based texture compression schemes such as DXTC. *Like DXTC the basic compression algorithm implemented in a preferred embodiment divides or splits the image into discrete blocks of pixels, (e.g., 4x4 pixel of 8x8 pixel blocks) and compresses them.*

However, rather than mechanistically compressing the regular texture blocks, the present invention introduces the idea of further subdividing or partitioning the block internally into “subsets” of different shape and size before compression.

Id. at 5:66-6:9 (emphasis added).

60. For example, Figure 8 illustrates a texture image that has been divided into 16 different 4x4 blocks 802, with each texture block being further partitioned into three disjoint subsets. One subset comprises the texels labeled 0, another subset comprises the texels labeled 1, and another comprises the texels labeled 2:

0, 0, 1, 1, 0, 0, 1, 1, 0, 2, 2, 1, 2, 2, 2, 2,	0, 0, 0, 1, 0, 0, 1, 1, 2, 2, 1, 1, 2, 2, 2, 1	0, 0, 0, 0, 2, 0, 0, 1, 2, 2, 1, 1, 2, 2, 1, 1,	2, 0, 0, 0, 2, 2, 0, 0, 2, 2, 1, 1, 2, 1, 1, 1,
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 1, 1, 2, 2,	0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 2, 2, 0, 0, 2, 2,	1, 1, 2, 2, 1, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0,	1, 1, 0, 0, 1, 1, 0, 0, 2, 2, 0, 0, 2, 2, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 2, 1, 2, 2,	0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2,	0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2,	0, 0, 1, 2, 0, 0, 1, 2, 0, 0, 1, 2, 0, 0, 0, 2,
0, 1, 1, 2, 0, 1, 1, 2, 0, 1, 1, 2, 0, 1, 1, 2,	0, 1, 2, 2, 0, 1, 2, 2, 0, 1, 2, 2, 0, 1, 2, 2,	0, 0, 1, 1, 0, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 2,	0, 0, 1, 1, 2, 0, 0, 1, 2, 2, 0, 0, 2, 2, 2, 0,

802

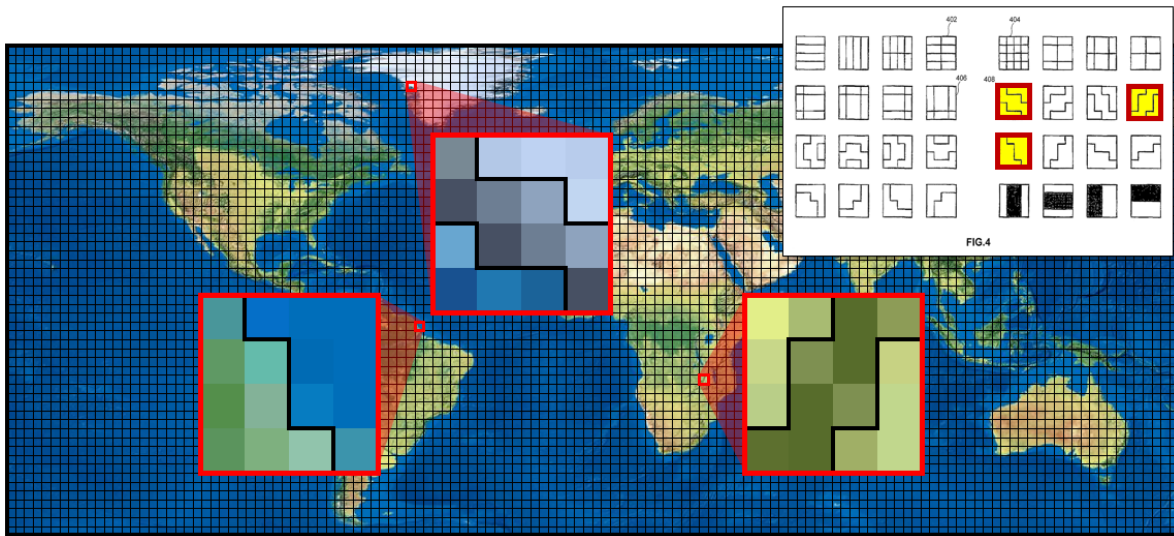
FIG. 8

Id. at Fig. 8, 6:60-7:15.

61. This subset approach addresses several image quality problems associated with the prior block-based schemes, resulting in “significant overall increase in perceived and measured quality. Furthermore, it provides improvement in the ability to compress less common-types of texture data (such as non-photorealistic textures, cartoons, etc.).” *Id.* at 6:10-16.

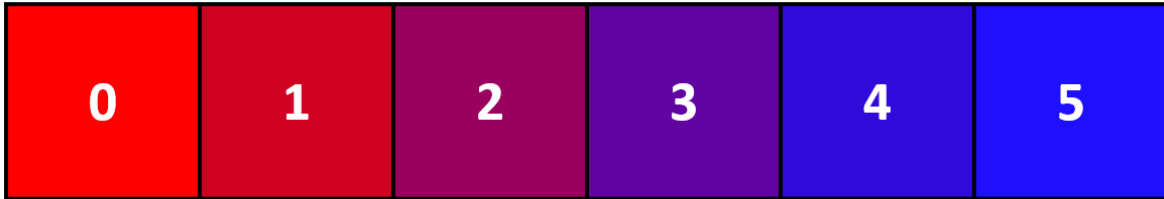
62. Below is an example of the use of the disjoint subsets of the ’628 Patent. The ability to match the color contours of an image using disjoint subsets is well illustrated by the block of pixels on the borders of Brazil and Greenland, where the colors transition from land to ocean. For example, with Greenland, the three

partitions (*i.e.*, disjoint subsets) match the three different groups of colors (blue, grey, and near white), thus reducing the compression error.

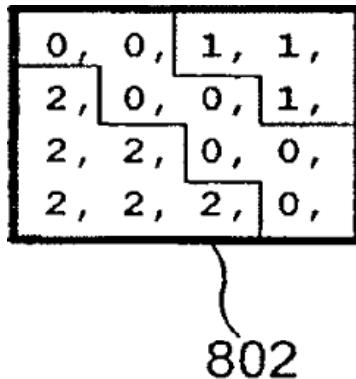


2. Index Values

63. Each block and its subsets are associated with certain information to allow them to be decompressed and used in texture mapping as necessary. *E.g.*, Ex. 1001 at 9:7-10:52. For example, to allow the color of a texel to be reconstructed during decompression, each subset is associated with (among other things) its own color endpoints. *E.g.*, *id.* at 10:61-11:10. These color endpoints were generated for the subset by using the two explicit colors as the endpoints of a line (0 and 5 in the image below) in a RGB color space, with the remaining colors (1, 2, 3, and 4 in the image below) being evenly distributed along that line. *Id.* at 10:61-64. The interpolation is simple linear interpolation. For example, color 1 below is the sum of 4/5 of color 0 and 1/5 of color 5.



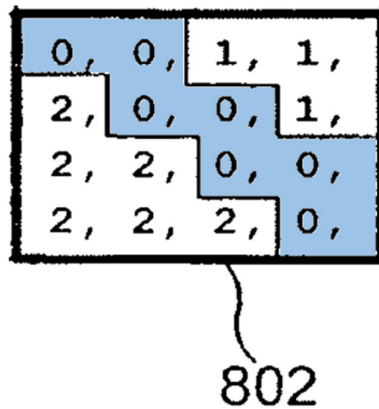
64. To illustrate, and as described in the '628 Patent, where a block is divided into three subsets (as described above in connection with Figure 8 and excerpted below), six color endpoints will be stored: two color endpoints for subset 0, two different color endpoints for subset 1, and two different color endpoints for subset 2. *Id.* at 11:50-54.



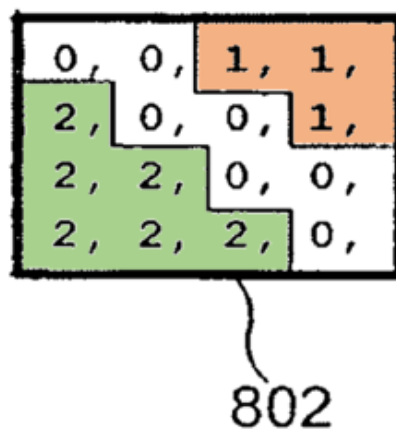
Id. at 10:32-36; 10:53-60; 9:40-57; Fig. 8.

65. Each texel in a given subset has its own index value associated with it, thereby allowing its color to be calculated based on color endpoints that are specific to that subset. *Id.* at 10:67-11:1. Specifically, each texel's index value indicates how to interpolate between the two color endpoints to generate a specific color for that texel. *Id.* at 10:61-11:1, 11:50-12:17. To continue our example, each of the

seven texels in subset 0 is represented by its own index value (a total of seven index values) indicating how to interpolate between the two color endpoints associated with subset 0. *Id.*



66. Similarly, each of the three texels in subset 1 (colored in orange) and each of the six texels in subset 2 (colored in green) is represented by its own index value (a total of three index values and six index values, respectively) indicating how to interpolate between the two color endpoints associated with each respective subset. *Id.*



67. This decompression approach, along with further dividing texture blocks into subsets as described above, “gives a very significant improvement in visual quality” over DXTC, for example, “noticeable improvement in perceived quality” and the elimination of image artifacts. *E.g.*, Ex. 1001 at 12:51-56. The image of the world in paragraph **Error! Reference source not found.** is a good example, where the techniques of the ’628 Patent enable better compression by matching the color contours of the texture image. Rectangular blocks cannot match the curved contours of the coasts that I have highlighted.

IX. DESCRIPTION OF ASSERTED ART

A. Sorgard (Ex. 1004)

68. “Sorgard” is a U.S. Patent No. 8,102,402 entitled “Method of and Apparatus for Encoding Data.”

69. Sorgard describes a compression and decompression scheme where the texture image is divided into blocks of 8x4 texels. Ex. 1004 at 24:23-38, Fig. 1. Each block is further divided into two blocks of 4x4 texels as shown below. *Id.*

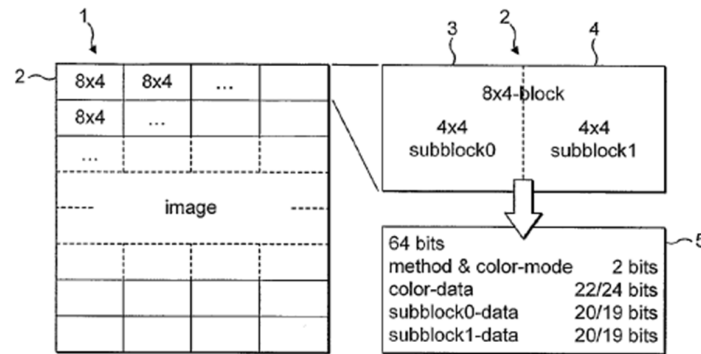
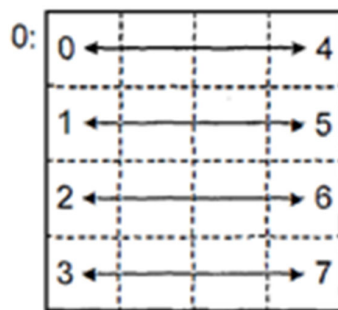


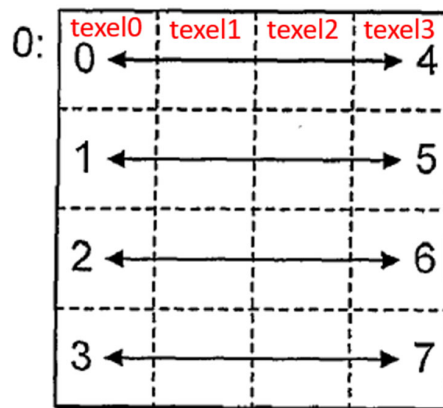
FIG. 1

70. Sorgard describes using one of eight “fading patterns” to represent and calculate the colors of texels in a 4x4 block. Each of Sorgard’s fading patterns “comprises a set of four interpolation paths that each individually cover four texture elements.” *Id.* at 33:24-28. For example, fading pattern 0 below comprises four interpolation paths (path 0-4, path 1-5, path 2-6, and path 3-7), each running horizontally from left to right and each comprising four texture elements (texels) represented by one of sixteen squares:



Id. at Fig. 5.

71. Sorgard labels the four texels within each interpolation path (from left to right) as texel0, texel1, texel2, and texel3. The texels at the endpoints of the path are texel0 and texel3 (as annotated below using path 0-4). The interior texels (the ones between the endpoints) are texel1, and texel2.



Id. at Fig. 5 (annotated).

72. To determine the colors of each texel in a path during decompression, Sorgard first describes that there is a color associated with each endpoint of a path, with $\text{color}[x]$ assigned to texel0 and $\text{color}[y]$ assigned to texel3. Then, to determine the colors of the interior texels, a preset formula is used: the color of interior texel1 is always $5/8$ of $\text{color}[x]$ plus $3/8$ of $\text{color}[y]$, while the color of interior texel2 is always $3/8$ of $\text{color}[x]$ plus $5/8$ of $\text{color}[y]$. Sorgard sets forth these color formulas as follows:

$$\begin{bmatrix} \text{texel0: } endpoint_{color}[x] \\ \text{texel1: } \left(endpoint_{color}[x] * \frac{5}{8} \right) + \left(endpoint_{color}[y] * \frac{3}{8} \right) \\ \text{texel2: } \left(endpoint_{color}[x] * \frac{3}{8} \right) + \left(endpoint_{color}[y] * \frac{5}{8} \right) \\ \text{texel3: } endpoint_{color}[y] \end{bmatrix}$$

Id. at 34:1-8. The intermediate colors of Sorgard are always interpolated using fixed weights of 3/8 and 5/8. This contrasts with the '628 Patent, which includes indices for each texel in the block to enable individual selection of color for each texel, including the intermediate texels. Ex. 1001 at 10:61-11:1, 11:50-12:17.

73. Sorgard also describes that each 4x4 block includes an index to choose one of the fading patterns. These different fading patterns are labeled as number 0 through 7 and are illustrated below:

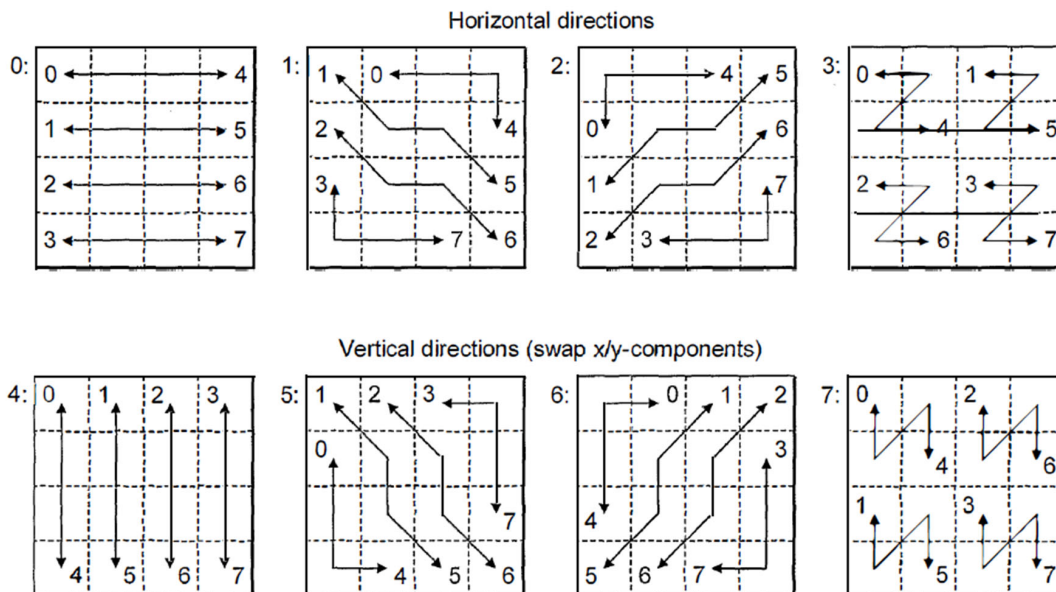


FIG. 5

Ex. 1004 at Fig. 5. Each fading pattern is defined by two major directions, horizontal and vertical, and four sub-directions. *Id.* at 33:29-35. The direction of the fading pattern is encoded into each block, with one bit used to indicate the major direction and two-bits to indicate the sub-direction. *Id.* at 33:37-47.

B. FXT1 (Ex. 1005)

74. “FXT1” is a whitepaper entitled “FXT1 Texture Compression Technology White Paper” describing at a very high level a block-based texture compression scheme. This is the same FXT1 approach described and distinguished in the ’628 Patent. *E.g.*, Ex. 1001 at 2:46-52, 2:67-3:9, 4:41-58; 5:66-6:1.

75. FXT1 explicitly describes dividing an entire image into blocks: “FXT1TM texture compression works by dividing an image into multiple 4x4 and/or 4x8 texel blocks.” Ex. 1005 at 5. These blocks are further encoded using one of four algorithms: CC_MIXED, CC_HI, CC_CHROMA or CC_ALPHA. *Id.* at 5-6. Dr. Manocha asserts that only two of these algorithms are relevant: CC_MIXED and CC_HI. Ex. 1003, ¶ 60.

76. **CC_MIXED** is similar to DXTC. It uses texture blocks of 4x4 texels and each block contains data for two colors. Ex. 1005 at 5. A four-entry color table is created using the two stored colors and two new ones interpolated from them. *Id.* A 2-bit index for each texel is used to determine which of the four colors to assign to each texel in the 4x4 block. *Id.*

77. **CC_HI** is similar to **CC_MIXED**, but uses texture blocks of 4x8 texels. *Id.* An eight-entry color table is created with two table entries populated by two colors stored in the block, five colors interpolated from the stored two colors, and an eighth color defined to represent transparency. *Id.* The block contains a 3-bit index for each texel, which is used to index one of the eight colors. *Id.* Therefore, **CC_HI** is simply an expanded version of DXTC.

X. EVALUATION OF PETITIONER'S PROPOSED GROUNDS

A. Ground 1: Sorgard and the Knowledge of a POSITA Does Not Render Claims 7-12 of the '628 Patent Obvious

78. In my opinion, claims 7-12 are not obvious in view of Sorgard and the knowledge of a POSITA.

1. Sorgard's Fading Pattern Index Is Not Either of the Claimed Index Values

79. Dr. Manocha opines that Sorgard teaches:

determining a first texel color for a first texel that is part of the first disjoint subset by interpolating between color endpoints of the first set of color endpoints based on a first index value for the first texel; and

determining a second texel color for a second texel that is part of the second disjoint subset by interpolating between color endpoints of the second set of color endpoints based on a second index value for the second texel.

Ex. 1001, Claim 7; Ex. 1003, ¶¶ 96-102. He identifies the below formula from Sorgard as meeting the above claim elements:

$$\begin{bmatrix} \text{texel0: } endpoint_{color}[x] \\ \text{texel1: } \left(endpoint_{color}[x] * \frac{5}{8} \right) + \left(endpoint_{color}[y] * \frac{3}{8} \right) \\ \text{texel2: } \left(endpoint_{color}[x] * \frac{3}{8} \right) + \left(endpoint_{color}[y] * \frac{5}{8} \right) \\ \text{texel3: } endpoint_{color}[y] \end{bmatrix}$$

Ex. 1004 at 34:1-8; Ex. 1003, ¶ 97. Specifically, Dr. Manocha asserts that the calculations for texel1 and texel2 discloses interpolating between color endpoints based on an index value. *Id.* I disagree with Dr. Manocha.

80. In my opinion, Sorgard does not disclose using an index value to interpolate between color endpoints for intermediate texels. Rather, as shown above, Sorgard uses a fixed linear equation that does not depend on Dr. Manocha's asserted index value. This linear equation provides a predetermined and constant amount of interpolation for any given texel between the color endpoints for a given fading pattern: "The colours for the *two remaining texture elements* in the middle of each interpolation path *are generated by doing linear interpolation* between each of the endpoint colours of the interpolation path. Thus, *in all cases* (any fading direction or pattern), *the resulting colours for the 4 texture elements covered by a given interpolation path will be:*

$$\begin{bmatrix} \text{texel0: } endpoint_{color}[x] \\ \text{texel1: } \left(endpoint_{color}[x] * \frac{5}{8} \right) + \left(endpoint_{color}[y] * \frac{3}{8} \right) \\ \text{texel2: } \left(endpoint_{color}[x] * \frac{3}{8} \right) + \left(endpoint_{color}[y] * \frac{5}{8} \right) \\ \text{texel3: } endpoint_{color}[y] \end{bmatrix}$$

Ex. 1004 at 33:62-34:5 (emphasis added and annotated). In other words, Sorgard’s method of determining the colors for intermediate texels is by interpolating between color endpoints exclusively based on a fixed formula that blends color endpoints at a 3/8 and 5/8 ratio. This method is not based on index values that are specific to a first and second texel.

81. Dr. Manocha identifies 16:10-17 of Sorgard as disclosing one of the claimed index values. Ex. 1003, ¶ 98. This part of Sorgard provides:

In another arrangement, only the data values for a selected number of the texture elements that the block encodes are indicated (e.g., the “endpoint” texture elements, where a fading pattern or interpolation path is to be used to derive data values for individual texture elements). This data could again be provided in the form of a data value index for each such texture element, or at least allow such an index to be derived.

Ex. 1004 at 16:10-17. I disagree with Dr. Manocha that this discloses either of the claimed “index value[s].” Sorgard’s fading pattern index value is not used to interpolate between color endpoints for intermediate texels. Rather, as explicitly

stated above, this index value only provides the fading pattern or interpolation direction.

82. Dr. Manocha also opines that Sorgard's fading pattern index value (indicated in green below) determines the interpolation formula used to derive its color. Ex. 1003, ¶ 99. That is incorrect. As explained above, Sorgard uses the same formula regardless of fading pattern – a fixed formula that blends color endpoints at a 3/8 and 5/8 ratio.

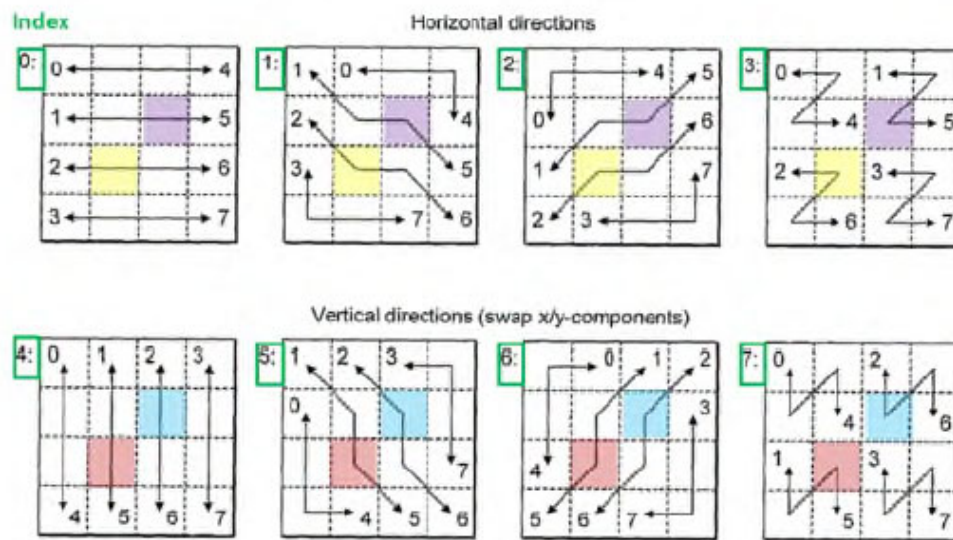


FIG. 5

Ex. 1003, ¶ 99.

83. Dr. Manocha's examples further show this is not true. Specifically, he opines that index values 0, 1, 2, or 3 (shown in green boxes above) “determines that the texel coordinate (1, 2) (yellow) is to be interpolated between endpointcolor[2] and [6], with endpointcolor[2] weighed more heavily, while the texel at coordinate

(2,1) (purple) is to be interpolated as between endpointcolor[5] and [1], with endpointcolor[5] weighed more heavily.” However, as Dr. Manocha admits, *for all four fading pattern indexes* (0, 1, 2, and 3), the yellow and purple texels are calculated the same way. This further shows that the specific fading pattern index has no bearing on the interpolation of intermediate texels.

84. Dr. Manocha opines that “[a]dditionally, or alternatively, Sorgard disclosed an index that is used to determine the direction of the fading pattern or interpolation path.” Ex. 1003, ¶ 101 (citing Ex. 1004 at 33:43-47). This is the same index value discussed in Sorgard at 16:10-17, which identifies an index indicating a fading pattern or interpolation path. Therefore, this is not a second, separate index value.

2. Dr. Manocha Fails to Identify a Second Index Value

85. Dr. Manocha does not identify a second index value, or identify which index value discloses “determining a second texel color for a second texel that is part of the second disjoint subset by interpolating between color endpoints of the second set of color endpoints based on a second index value for the second texel.” Ex. 1003, ¶¶ 102-103. He simply opines that that “the reasons above in claim limitation 7[d] apply to this claim limitation, with the addition that Sorgard’s decompression algorithm is repeated for a second texel that is part of a second disjoint subset, which a POSITA would have understood to be inherent in Sorgard.” *Id.*

86. Dr. Manocha identifies Sorgard's 4x4 subblock as the claimed "compressed texture block," and each interpolation path as a "disjoint subset." Ex. 1003, ¶ 83 ("Sorgard comprises a compressed texture block including two or more disjoint subsets. Specifically, Sorgard disclosed *dividing each 4x4 texture block into a plurality of interpolation paths.*") (emphasis added). According to this premise, in order to disclose the required index values, at least two of the interpolation paths (*i.e.*, the alleged disjoint subset) must each have an associated index value for "determining a . . . texel color . . . by interpolating between color endpoints of the . . . set of color endpoints based on a[n] . . . index value."

87. Additionally, the plain and ordinary meaning of the claims require that the first and second texels are interpolated using two different index values, each specific to its respective texel: "a first index value for the first texel," and "a second index value for the second texel." Ex. 1001 at Claim 7.

88. However, Sorgard provides that one index value applies to the entirety of the 4x4 subblock, as it must because the single fading pattern governs the entirety of a 4x4 subblock:

In order to encode the fading directions to be used with this texture element colour generation method, the direction of the fade pattern is identified and encoded in the texture data blocks. This encoding consists of including in the texture data block the colour at each end point in the fade pattern and the direction of the fade pattern (*i.e.* which major direction, and which sub-direction, as shown in FIG. 5, is to be used). For this encoding, one bit is used to indicate the major direction

of the fading pattern, and two bits are used to indicate the sub-direction (i.e. *as an index* into which of the four sub-directions is to be used).”

Ex. 1004 at 33:37-47 (emphasis added); *id.* at Fig. 5. While the fading pattern index value has two components to it, one that indicates the direction (horizontal or vertical) of the fading pattern, and another that indicates the sub-direction of the fading pattern, both components work together to identify the single fading pattern applicable to the entire 4x4 subblock. Ex. 1004 at 33:43-47, 33:1-22, Fig. 5.

89. Therefore, in my opinion, Sorgard discloses only one index value as applying to the entire block, and not different index values that are respective of different texels.

B. Ground 2 and 4⁵: FXT1 and the Knowledge of a POSITA Does Not Render Claims 7-12 Obvious, Nor Does the Addition of Reshetov

90. In my opinion, claims 7-12 are not obvious in view of FXT1 and the knowledge of a POSITA as alleged in Ground 2, nor are they obvious with the addition of Reshetov as alleged in Ground 4.

1. Construction of “Block” and “Subset”

91. Dr. Manocha claims that the plain and ordinary meaning of the term “compressed texture block” “can read on a block comprising an entire image or a

⁵ The Petition does not include a Ground 3, although Dr. Manocha’s Declaration refers to Ground 4 as Ground 3.

part of an image, and the claim does not specify a size of the block.” Ex. 1003, ¶ 140. I disagree. As explained further below, defining “compressed texture block” to encompass the entirety of a texture or image is too broad, and incorrect.

92. The term “block” is a well-understood term in the industry. A POSITA would understand the term “block” to mean “a small group of pixels comprising a division of a full image, but not the full image itself.” As explained above, large textures were divided into blocks to improve texture fetching and enable storage in the texture cache of the GPU. A POSITA would not consider the entire texture image to be a block because loading a complete image would defeat the purpose of using block-based texture compression, which is to load small chunks of texture into a GPU cache to conserve bandwidth and increase performance.

93. Hakura and Gupta wrote the seminal paper on GPU texture caching. Ex. 2004. Below is a block diagram of their proposed texture cache architecture, which is still in use today. *Id.* at 109. A fast static random access memory (SRAM) is used as the cache to retrieve portions of texture images from the dynamic random access memory (DRAM). On GPU cards, the SRAM is on the GPU chip, while DRAM is on external chips. This is just like the conventional architecture that is used on CPUs to retrieve instructions and data.

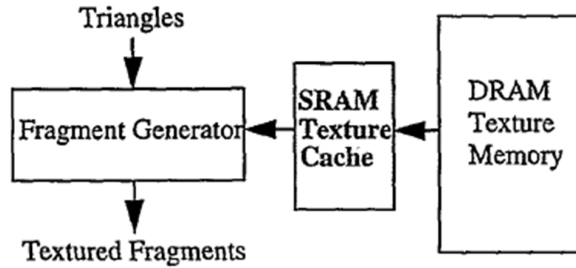


FIGURE 2.1. Block diagram of texture mapping hardware.

94. As Hakura and Gupta point out, DRAM is designed for burst access (*i.e.* a block of data is the chunk that is transferred). As they write, “block transfers of cache lines between the cache and memory make it possible to get the most bandwidth out of the memory.” Ex. 2004 at 110.

95. They write that “static memories typically have shorter access latencies than dynamic memories. The SRAM cache can be tightly coupled with the fragment generator which makes it possible to run the fragment generator at higher clock rates. This is an incentive for integrating the SRAM cache onto the same chip as the fragment generator.” Ex. 2004 at 110.

96. Hakura and Gupta advocate a “blocked representation.” Ex. 2004 at 113. As they write, “[a] blocked representation can be used to reduce the dependency on the orientation of textures, as seen by the viewer, and to exploit more spatial locality.” *Id.* Furthermore, they write “[w]e can conclude that the blocked representation is an essential component for reducing the frequency of capacity misses” and “we can conclude that the best block size to use for texturing

corresponds to when the memory required to store one block of texture is the same as the cache line size.” Ex. 2004 at 114. A POSITA would have known that a cache line is small, and certainly not the size of an image. For example, Hakura generated experimental data with block sizes of 2x2, 4x4, 8x8, and 16x16. *Id.*

97. Additionally, the ’628 Patent specification states, “the present invention represents a further refinement of block-based texture compression schemes such as DXTC. Like DXTC, the basic compression algorithm implemented in a preferred embodiment *divides or splits the image up into discrete blocks of pixels* (e.g., 4×4 pixel or 8×8 pixel blocks) and compresses them.” Ex. 1001 at 5:66-6:4 (emphasis added). As this quotation also demonstrates, the invention of the ’628 Patent extends prior block-based image compression techniques, for example, DXTC and FXT1. *Id.* The ’628 Patent’s usage of “block” comports with how “block” is used in Hakura, and also in DXTC (also called S3TC, e.g., Ex. 2006): “The *image* is broken up into 4x4 *blocks*.”). Ex. 2007 at 1 (emphasis added).

98. Konstantine I. Iourcha, the first-named inventor on the ’628 Patent, is also credited as having invented DXTC. *E.g.*, Ex. 2008 at 2 (Hua) (“Developed by Iourcha and his collaborators [7], S3 Texture Compression (or called DXTC in Microsoft DirectX 3D) is a lossy texture compression algorithm commonly used in commercial graphics hardware.”).

99. FXT1 also uses “block” the same way as DXTC and the ’628 Patent (and as the industry does): “FXT1™ texture compression works by *dividing an image into multiple 4x4 and/or 4x8 texel blocks.*” Ex. 1005 at 5 (emphasis added). The ’628 Patent also states that FXT1 is an extension of DXTC, further confirming that its usage of “block” is the same. Ex. 1001 at 2:46-50 (“FXT 1 is a competing compression scheme with DXTC. It essentially extends DXTC with some additional block types that can be mixed within any given image, and also provides a 4 bits per pixel compression mode for textures so with alpha.”).

100. Reshetov, relied on by Dr. Manocha, also distinguishes between a complete image/texture and a block. It defines “texture,” which “is an image” and “texture blocks,” which are “[p]ortions of a texture”:

“Texture: A texture is an image that is used to enhance the appearance of surfaces in three-dimensional graphics. Textures may be one-dimensional, two-dimensional or multidimensional textures, including MIPS maps.”

Ex. 1008 at [0048].

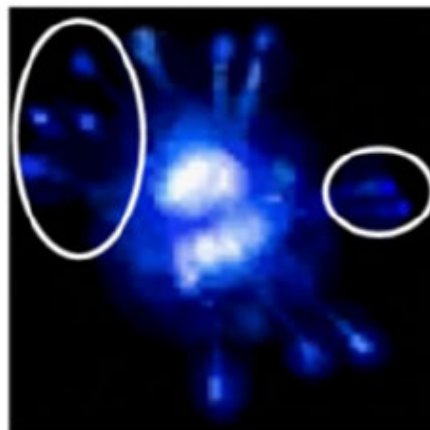
“Texture Blocks: Portions of a texture that contain a plurality of texels. Texture blocks may be overlapping texture blocks and non-overlapping (two texture blocks that do not have any texels in common) texture blocks. Typically texture blocks are square or rectangular. Typically, texture blocks are also naturally aligned meaning that the texture blocks are aligned along rows and columns. For example, in one embodiment a texture block comprises a 4×4 block of non-overlapping texels.”

Id. at [0049].

101. The term “subset” is also a well-understood term in the industry. A POSITA would define “subset” as “a subdivision or partition of a block.” This meaning is also the meaning that the ’628 Patent ascribes to the term “subset.” The ’628 Patent specification states: “[R]ather than mechanistically compressing the regular texture blocks, the present invention introduces the idea of *further subdividing or partitioning the block internally into ‘subsets’* of different shape and size before compression.” Ex. 1001 at 6:5-9 (emphasis added).

2. Under the Correct Construction, Dr. Manocha Fails to Allege that FXT1 Discloses “Subsets”

102. Dr. Manocha opines that “[a] POSITA would have understood that the compression algorithm outputs *a compressed image comprising a compressed texture block.*” Ex. 1003, ¶ 133 (emphasis added). Dr. Manocha further opines that the below graphic is an “image represented by a texture block.” *Id.* (citing Ex. 1005 at 7).



But FXT1 refers to the above graphic not as a texture block, but as an *image*. Ex. 1005 at 7 (“Uncompressed image (left) compared to the same image compressed using S3TC (center) compared to FXT1 (right).”). Dr. Manocha thus opines that an entire image satisfies the claim requirement of a “block,” which is incorrect, as explained above.

103. As to “subset,” Dr. Manocha argues that “FXT1 discloses that data in the compressed texture block is to be unpacked into the two or more disjoint subsets” because “FXT1 compressed data are partition[ed] into 4-textel-by-4-textel or 4-textel-by-8-textel disjoint subsets.” Ex. 1003, ¶ 138. This is also incorrect. What FXT1 actually provides is that those 4x4 or 4x8 pixel subdivision are the first level of subdivisions of the whole image, *i.e.*, they are the blocks. FXT1 even uses the word “blocks” to describe them: “FXT1™ texture compression works by *dividing an image into multiple 4x4 and/or 4x8 texel blocks*.” Ex. 1005 at 5 (emphasis added).

104. Dr. Manocha needed to identify a further subdivision of the first-level subdivision of the image in order to meet the requirement of a “subset,” but, because no such disclosure exists in FXT1, has failed to do so.

3. Petitioner’s Suggested Modifications Would Render FXT1 Inoperable

105. In my opinion, Dr. Manocha’s proposed modification—which is that instead of operating on the pixels of one subdivision (*i.e.*, “block”) of an image at a

time before moving onto the next subdivision, FXT1 could operate on all pixels of the whole image at once, and thus could consider the whole image a “block,” Ex. 1003, ¶ 140 —would render FXT1’s system inoperable.

106. As I explained above in Sections VII.C.2 and X.B.1, there are real constraints in graphics processing that prevent operating on an entire image at once. That is one of the fundamental reasons that block-based compression algorithms exist: otherwise, all graphics processing would simply be done on the entire image, eliminating an unnecessary step of dividing the image into smaller blocks.

107. To explain further, GPUs are designed to load a portion of a texture image, for example, a 4x4 texels block, and not to load the entire image, for example, 2048x2048 texels. This is consistent with FXT1’s disclosure, which is designed to handle 4x4 or 4x8 texel blocks. Ex. 1005 at 2 (“As a free, open-source, cross-platform compression algorithm, 3dfx’s FXT1™ texture compression technology allows content developers to create textures at higher resolutions and use more textures in a given scene while simultaneously reducing the memory bandwidth required for texturing.”), 5 (“FXT1™ texture compression works by dividing an image into multiple 4x4 and/or 4x8 texel blocks.”)

108. Attempting to load full-size textures instead of the modestly-sized blocks for which FXT1 was designed would exceed the internal memory capacity (the texture cache) in the GPU, rendering the GPU inoperable. Even if one were

able to modify the GPU to make the vast internal memory capacity increases that would be necessary to handle the whole image, this would in turn require much larger die sizes and much greater expense. Even then, without making further significant changes to the microarchitecture and data paths, the system would not work, or would work very slowly. For example, a 1Kx1K image compressed at 4 bits per texel, as claimed in FXT1, would consume 524,288 bytes (or 512 KB) of memory space in contrast to a 4x4 block that consumes 8 bytes. *Id.* at 2. As a reminder, level 1 caches at the time stored less than 12 KB, so the full texture could not even be stored. *See* Ex. 2005 (Doggett) at 139; *see supra* Section VII.C.2. This is why everyone in the industry uses small blocks. Also, keep in mind that multiple textures are used at once, so memory would be required for many textures, not just one. This is simply impractical and would render a system using such a modification of FXT1 inoperable.

4. Dr. Manocha's Other Proposed Modification to FXT1 is Hindsight Without Any Motivation

109. Dr. Manocha also opines that one could modify FXT1 by adding another division of the image at a level between the whole image and what FXT1 calls a block. Ex. 1003, ¶ 141. My understanding is that one could call the new intermediate subdivision a “block” and call what FXT1 calls a block a “subset.” In my opinion, this is contrary to the method disclosed by FXT1.

110. FXT1 provides no reason to create this additional theoretical level of subdivisions that conveniently divide back down into the blocks described in FXT1. And Dr. Manocha does not say how this would improve the functionality of FXT1. He simply states that “[a] POSITA would have been motivated to make an additional level of partitioning to, for example, break up a large image into sets of data such that performing compression on each block would preserve the image fidelity of the texture” but does not say why or how the fidelity would be preserved any differently than without the extra level of subdivision. Ex. 1003, ¶ 143.

111. Dr. Manocha does not tell us whether or how the compression algorithms of FXT1 would be affected by this theoretical subdivision. A POSITA would not be motivated to add an intermediary level of subdivisions because they would not increase the quality of the compression. The rectangular subdivisions proposed by Dr. Manocha would not solve the problems addressed by the ’628 Patent, which indicates that the subsets of the ’628 Patent can be “of different shape and size.” Ex. 1001 at 6:8-9. For example, rectangular subsets would not adequately match the color gradients of the coasts of Brazil or Greenland in the example of paragraph 62 above.

XI. CONCLUSION

112. 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.

113. 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.

114. 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.

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

July 26, 2023

Respectfully Submitted,



Dr. Anselmo Lastra

EXHIBIT A

CURRICULUM VITAE

Anselmo A. Lastra

ADDRESS: 1407 Ruffin Street
Durham, NC 27701
alastra@gmail.com
lastra@cs.unc.edu
<http://www.cs.unc.edu/~lastra>

(919) 260-9772

CURRENT EMPLOYER: none

EDUCATION:

1988 Duke University, Durham, NC - Ph.D. in Computer Science
M.A. in Computer Science from Duke
1972 BSEE from Georgia Institute of Technology

PROFESSIONAL EXPERIENCE:

2017- Professor Emeritus, University of North Carolina at Chapel Hill.

2009-2014 Department Chair

2006-2017 Professor of Computer Science, and Research Associate of the Research Laboratories of Archaeology, University of North Carolina. Engaged in research in the area of computer graphics, principally rendering, image-based rendering and modeling, and hardware.

2006-2009 Director of Graduate Studies

2001-2006 Associate Professor of Computer Science, University of North Carolina.

1997-2001 University of North Carolina. Research Associate Professor.

1991-1997 University of North Carolina at Chapel Hill. Research Assistant Professor of Computer Science. Software Manager for the Pixel-Planes team. Over this time we completed the Pixel-Planes 5 and (with colleagues at Hewlett Packard) PixelFlow parallel graphics computers.

1988-1991 Duke University. Research Assistant Professor of Computer Science. Research in the areas of parallel processing and performance, and computational science.

Updated July 3, 2023

1

Working on problems in non-linear excitable media, and the mathematical aspects of genetics. Interests in graphics, and pattern recognition.

1986-1988 Duke University. Graduate student in the Department of Computer Science. Dissertation research on parallel algorithms for the simulation of biological processes at the cellular level. Related research topics in the prediction of the performance of parallel algorithms, and debugging of parallel programs.

1986 AT & T Bell Laboratories. Consultant for one year on a project to develop a special purpose dataflow machine for signal processing. Responsible for technical supervision of several people as well as software design.

1979-1985 Duke University and Duke University Medical Center. Research Assistant in the Department of Medicine and graduate student in the Department of Computer Science. Research in pattern recognition, both statistical and structural. Developed two systems for automated recognition and classification of electrocardiograms. Work in graphics, UNIX kernel, and embedded systems.

Previous Coulter Electronics, Inc., Hialeah, Florida. Project Engineer. Managed design and development team of six to ten persons involved in hardware and software design of laboratory instrumentation. Supervised one product, the Coulter Lytning, completely from initial design to production.

Scidata, Inc., Atlanta, Ga. Electronic Engineer.

PUBLICATIONS

BOOK CHAPTER

Lastra, Anselmo, "The Unix Programming Environment" in *Mastering Tools, Taming Daemons: Unix for the Wizard Apprentice*, D. Brock, ed., McGraw Hill, 1995, ISBN: 1-884777-07-4, 69-110.

REFEREED

Lincoln, Peter, Andrei State, Turner Whitted, Anselmo Lastra, and Henry Fuchs, "From Motion to Photons in 80 Microseconds: Towards Minimal Latency for Virtual and Augmented Reality," IEEE VR, to appear, 2016.

Feng Zheng, Andrei State, Andrew Maimone, Turner Whitted, Anselmo Lastra, Peter Lincoln, and Henry Fuchs, "Minimizing Latency for Augmented Reality Displays: Frames Considered Harmful," ISMAR 2014.

Jeff Pool, Anselmo Lastra, and Montek Singh, Lossless Compression of Variable-Precision Floating-Point Buffers on GPUs, Symposium on Interactive 3D Graphics, Costa Mesa, CA, 9-11 March 2012.

Jeff Pool, Anselmo Lastra, and Montek Singh, Precision Selection for Energy-Efficient Pixel Shaders, High Performance Graphics 2011, Vancouver, 5-7 August 2011.

Jeff Pool, Anselmo Lastra, and Montek Singh, Power-Gated Arithmetic Circuits for Energy-Precision Tradeoffs in Mobile Graphics Processing Units, J. Low Power Electron. 7, 148-162, 2011.

Jeff Pool, Anselmo Lastra and Montek Singh, An Energy Model for Graphics Processing Units, IEEE International Conference on Computer Design, 2010.

Amela Sadagic, Greg Welch, Chumki Basu, Chris Darken, Rakesh Kumar, Henry Fuchs, Hui Cheng, Jan-Michael Frahm, Mathias Kolsch, Neil Rowe, Herman Towles, Juan Wachs, and Anselmo Lastra. New Generation of Instrumented Ranges: Enabling Automated Performance Analysis. In Proceedings of 2009 Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC-2009), 2009.

Jeff Pool, Anselmo Lastra and Montek Singh, Energy-Precision Tradeoffs in Mobile Graphics Processing Units, IEEE International Conference on Computer Design, Lake Tahoe, 2008.

Joshua Steinhurst, Greg Coombe and Anselmo Lastra, Reducing Photon Mapping Bandwidth by Query Reordering, IEEE Transactions on Visualization and Computer Graphics, **(14)** 1, 13-24, 2008.

Kok-Lim Low and Anselmo Lastra, Predetermination of ICP Registration Errors and Its Application to View Planning, 3-D Digital Imaging and Modeling (3DIM), Montreal, Canada, August 2007.

Massimo Brizzi, Sarah Court, Ascanio d'Andrea, Anselmo Lastra and Daniele Sepio, 3D Laser Scanning as a Tool for Conservation: The Experiences of the Herculaneum Conservation Project, Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST), Nicosia, Cyprus, 2006.

Kok-Lim Low and Anselmo Lastra, An Adaptive Hierarchical Next-Best-View Algorithm for 3D Reconstruction of Indoor Scenes, Proceedings of 14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006), Taipei, Taiwan, 2006.

Joshua Steinhurst and Anselmo Lastra, Global Importance Sampling of Glossy Surfaces using the Photon Map, IEEE Symposium on Interactive Ray Tracing, Salt Lake City, Utah, 2006.

Chad Hantak and Anselmo Lastra, Metrics and Optimization Techniques for Registration of Color to Laser Range Scans, Symposium on 3D Data Processing, Visualization and Transmission (3DPVT), Chapel Hill, NC, 2006.

Greg Coombe and Anselmo Lastra, An Incremental Weighted Least Squares Approach to Surface Lights Fields, GRAPP 2006, International Conference on Computer Graphics Theory and Applications, Setúbal, Portugal, 25 - 28 February, 84-91, 2006.

Greg Welch, Ruigang Yang, Sascha Becker, Adrian Ilie, Dan Russo, Jesse Funaro, Andrei State, Kok-Lim Low, Anselmo Lastra, Herman Towles, Bruce Cairns, M.D., Henry Fuchs, and Andy van Dam. Immersive Electronic Books for Surgical Training. *IEEE Multimedia*, 22-35, July-Sept 2005.

David K. McAllister, Anselmo Lastra, Measuring and Rendering Spatially-Varying Fabrics, *Research Journal of Textile and Apparel*, (9) 1, 1-12, 2005.

Justin Hensley, Thorsten Scheuermann, Montek Singh, Anselmo Lastra, Fast Summed-Area Table Generation and its Applications, Proceedings of Eurographics 2005 and Computer Graphics Forum (24) 3, 547-555, 2005.

Justin Hensley, Montek Singh, and Anselmo Lastra, A Fast, Energy-Efficient Z-Comparator, Proceedings of Graphics Hardware 2005, Los Angeles, 41-44, 2005.

Greg Coombe, Chad Hantak, Radek Grzeszczuk, Anselmo Lastra, Online Construction of Surface Light Fields, Eurographics Symposium on Rendering, 83-90, 2005.

Josh Steinhurst, Greg Coombe, Anselmo Lastra, Reordering for Cache Conscious Photon Mapping, Proceedings of Graphics Interface 2005, May 9-11, Victoria, British Columbia, Canada, 97-104, 2005.

Justin Hensley, Anselmo Lastra, Montek Singh, A Scalable Counterflow-Pipelined Asynchronous Radix-4 Booth Multiplier, Proceedings of the 11th IEEE International Symposium on Asynchronous Circuits and Systems, IEEE Computer Society, Washington, DC, USA, 128-137, 2005.

Justin Hensley, Anselmo Lastra, Montek Singh: An Area- and Energy-Efficient Asynchronous Booth Multiplier for Mobile Devices. ICCD 2004: 18-25, IEEE Press, 2004.

Nathaniel Williams, Kok-Lim Low, Chad Hantak, Mark Pollefeys, and Anselmo Lastra, Automatic Image Alignment for 3D Environment Modeling, Proceedings of SIBGRAPI 2004, Curitiba, Brazil, October 17-20, 2004, 388-395.

Ilie, Adrian, Kok-Lim Low, Greg Welch and Anselmo Lastra, "Combining Head-Mounted and Projector-Based Displays for Surgical Training", *Presence: Teleoperators and Virtual Environments* 13(2), 128-145, MIT Press, 2004.

Greg Coombe, Mark J. Harris, and Anselmo Lastra, Radiosity on Graphics Hardware, Proceedings of Graphics Interface 2004, May 17-19, London, Ontario, Canada, 2004, AK Peters, 161-168, 2004.

Nathaniel Williams, Chad Hantak, Kok-Lim Low, John Thomas, Kurtis Keller, Lars Nyland, David Luebke, and Anselmo Lastra, Monticello Through the Window, Proceedings of the 4th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage (VAST2003), Brighton, UK (November 2003), 131-138.

Kok-Lim Low and Anselmo Lastra, Reliable and Rapidly-Converging ICP Algorithm Using Multiresolution Smoothing, 4th IEEE International Conference on 3-D Digital Imaging and Modeling (3DIM 2003), 171-178, October 2003.

Celso Setsuo Kurashima, Anselmo Lastra, Marcelo Knorich Zuffo, A Virtual Simulation Tool for Development of Image-based Rendering Algorithms, Proceedings of 6th SVR Symposium on Virtual Reality, SBC Brazilian Computer Society, October 15-18, 2003. pp.60-70.

Mark J. Harris, William Baxter, Thorsten Scheuermann, Anselmo Lastra, Simulation of Cloud Dynamics on Graphics Hardware, Graphics Hardware 2003, San Diego, California, July 26-27, 92-101, 2003.

Kok-Lim Low, Adrian Ilie, Greg Welch, and Anselmo Lastra, Combining Head-Mounted and Projector-Based Displays for Surgical Training, Proceedings of IEEE Virtual Reality 2003. Los Angeles, California, March 22-26, 2003.

Ruigang Yang, Celso Kurashima, Andrew Nashel, Herman Towles, Anselmo Lastra, Henry Fuchs, Creating Adaptive Views for Group Video Teleconferencing – An Image-Based Approach, Proceedings of International Workshop on Immersive Telepresence (ITP2002), Juan Les Pins, France, December 6, 2002.

Kurashima, Celso, Ruigang Yang, and Anselmo Lastra, Combining Approximate Geometry with View-Dependent Texture Mapping – A Hybrid Approach to 3D Video Teleconferencing, Proc. of the 15th Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI'02, Fortaleza, CE, Brazil, October 7-10, 2002

Harris, Mark J., Greg Coombe, Thorsten Scheuermann, and Anselmo Lastra, Physically-Based Visual Simulation on Graphics Hardware, Graphics Hardware 2002, Saarbrücken, Germany, September 1-2, 109-118.

McAllister, David K., Anselmo A. Lastra, and Wolfgang Heidrich, Efficient Rendering of Spatial Bi-directional Reflectance Distribution Functions, Graphics Hardware 2002, Saarbrücken, Germany, September 1-2, 79-88.

Kok-Lim Low, Greg Welch, Anselmo Lastra and Henry Fuchs, Life-Sized Projector-Based Dioramas, *ACM Symposium on Virtual Reality Software and Technology 2001 (VRST 2001)*, November 2001.

Harris, Mark, and Anselmo Lastra, Real-Time Cloud Rendering, Computer Graphics Forum (Eurographics 2001 Proceedings), 20(3):76-84, Manchester, UK, September 4-7, 2001.

Popescu, Voicu, Anselmo Lastra, The Vacuum Buffer, Proceedings of the 2001 *ACM Symposium on Interactive 3D Graphics*, Research Triangle Park, NC, March 19-21, 2001, 73-76.

Popescu, Voicu, Anselmo Lastra, John Eyles, Sort-First Parallelism for Image-Based Rendering, *Proceedings of Eurographics Workshop on Parallel Graphics and Visualization*, Girona, Spain, September 2000, 93-101.

Popescu, Voicu, John Eyles, Anselmo Lastra, Josh Steinhurst, Nick England and Lars Nyland, WarpEngine: An Architecture for the Post-Polygonal Age, *Proceedings of SIGGRAPH 2000*, New Orleans, July 2000, 433-442.

Lars Nyland, David McAllister, Voicu Popescu, Chris McCue, and Anselmo Lastra. Interactive exploration of acquired 3D data. Proceedings of the SPIE Applied Image and Pattern Recognition Conference, October 13-15, 1999

Aliaga, Daniel and Anselmo Lastra, Automatic Image Placement to Provide a Guaranteed Frame Rate, *Proceedings of SIGGRAPH 99*, Los Angeles, August 11-13, 1999, 307-316.

Chang, Chun-Fa, Gary Bishop, and Anselmo Lastra, LDI Tree: A Hierarchical Representation for Image-based Rendering, *Proceedings of SIGGRAPH 99*, Los Angeles, August 11-13, 1999, 291-298.

McAllister, David K., Lars Nyland, Voicu Popescu, Anselmo Lastra, Chris McCue. Real-Time Rendering of Real-World Environments, *Rendering Techniques '99, Proceedings of the Eurographics Workshop on Rendering*, (Granada, Spain), June 21-23, 1999.

Lars Nyland, David McAllister, Voicu Popescu, Chris McCue, Anselmo Lastra, Paul Rademacher, Manuel Oliveira, Gary Bishop, Gopi Meenakshisundaram, Matt Cutts, and Henry Fuchs, The Impact of Dense Range Data on Computer Graphics, *Proceedings of Multi-View Modeling and Analysis Workshop (MVIEW99), (Part of CVPR99)*, Fort Collins, CO, June 23-26, 1999.

Bastos, Rui, Kenneth Hoff, William Wynn, and Anselmo Lastra, "Increased Photorealism for Interactive Architectural Walkthroughs", *ACM Symposium on Interactive 3D Graphics*, Atlanta, April 26-28, 1999, pp. 183-190.

Rafferty Matthew M., Daniel G. Aliaga, Voicu Popescu, and Anselmo A. Lastra, "Images for Accelerating Architectural Walkthroughs", *IEEE Computer Graphics & Applications*, **18**:6, pp. 38-45, November/December 1998.

Voicu Popescu, Anselmo Lastra, Daniel Aliaga, Manuel Oliveira Neto, "Efficient Warping for Architectural Walkthroughs using Layered Depth Images", *IEEE Visualization '98*, pp. 211-215, October 18-23, 1998.

Aliaga, Daniel G. and Anselmo A. Lastra, "Smooth Transitions in Texture-based Simplification", *Computer & Graphics*, Elsevier Science, **22**:1, pp. 71-81, 1998.

Rafferty, Matthew M., Daniel G. Aliaga, and Anselmo A. Lastra, 3D Image Warping in Architectural Walkthroughs, *Proceedings of VRAIS 98, (also UNC Computer Science Technical Report TR97-019, September 1997)*.

Olano, Marc and Anselmo Lastra, "A Shading Language on Graphics Hardware: The PixelFlow Shading System", *Proceedings of SIGGRAPH 98*, pp. 159-168, Orlando, Florida, July 19-24, 1998.

Aliaga, Daniel. G. and Anselmo Lastra, Architectural Walkthroughs Using Portal Textures, *Proceedings of IEEE Visualization 97*, Phoenix, AZ, October 19-24, 1997, 355-362.

Eyles, John, Steve Molnar, John Poulton, Trey Greer, Anselmo Lastra, and Nick England, PixelFlow: The Realization, *Proceedings of the SIGGRAPH/Eurographics Workshop on Graphics Hardware*, Los Angeles, CA, August 3-4, 1997, 57-68.

Kumar, Subodh, Dinesh Manocha, and Anselmo Lastra, Interactive Display of Large NURBS Models, *IEEE Transactions on Visualization and Computer Graphics*, December 1996, 323-336.

Hughes, Merlin, Anselmo Lastra, and Edward Saxe, Simplification of Global-Illumination Meshes, *Proceedings of Eurographics '96*, Computer Graphics Forum, **15**(3), August 1996, 339-345.

Lastra, Anselmo, Reducing Latency on PixelFlow, *Proc. of the Eurographics Workshop on Graphics Hardware*, 43-50, August 1995.

Lastra, Anselmo, Steve Molnar, Marc Olano and Yulan Wang, Real-Time Programmable Shading, *Proc. of the 1995 Symposium on 3D Interactive Graphics, SIGGRAPH*, 1995, 59-66.

Kumar, Subodh, Dinesh Manocha, and Anselmo Lastra, Interactive Display of Large-Scale NURBS Models, *Proc. of the 1995 Symposium on 3D Interactive Graphics, SIGGRAPH*, 1995, 51-58.

Nesterenko, V. V., Lastra, A. A., Grant, A. O. Rosenshtraukh, L. V., Starmer, C. F. A Proarrhythmic Response to Sodium Channel Blockade: The Influence of Antiarrhythmic Drugs on the Window of Vulnerability in Guinea Pig Myocardium. *Journal of Clinical Pharmacology*, **19**, No. 5, 810-820 (1992).

Speer, M. C., Yamaoka, L. H., Gilchrist, J. M., Gaskell, C. P., Stajich, J. M., Vance, J. M., Kazantsev, A., Lastra, A. A., Haynes, C. S., Beckmann, J. S., Weber, J. L., Roses, A. D., Pericak-Vance, M. A., Confirmation of Genetic Heterogeneity in Limb-Girdle Muscular Dystrophy: Linkage of an Autosomal Dominant Form to Chromosome 5q. *American Journal of Human Genetics*, June 1992.

Starmer, C. F., Lancaster, A. R., Lastra, A. A., Grant, A. O., Cardiac Instability Amplified by Use-Dependent Na Channel Blockade, *American Journal of Physiology*, **262**, H1305-H1310, 1992.

Starmer, C. F., Lastra, A. A., Nesterenko, V. V. Grant A. O., A Proarrhythmic Response to Sodium Channel Blockade: Theoretical Model and Computer Simulations, *Circulation* **84**, No.3, 1364-1377 (1991).

Lastra, A. A., and Starmer, C. F., POET: A Tool for the Analysis of the Performance of Parallel Algorithms, *Proc. 1988 Int. Conf. on Parallel Processing*, Pennsylvania State University Press, University Park, 126-129, 1988.

Simpson, E. V., Lastra, A. A., Pritchett, E. L. C. and Smith, W. M., A time-sharing computer system for rhythm analysis of long-term electrocardiographic data. *Proc. 9th Ann. Conf. IEEE Eng. Med. Biol. Soc.* (1987) 914-915.

Lastra, A. A., Pritchett, E. L. C., Reiter, M. J., Smith, M. S., and Smith, W. M., A System for the Analysis of Long-Term Electrocardiographic Studies in Clinical Research and Training. *Computers and Biomedical Research* (1983) 340-346.

EXHIBITION

Virtual Monticello exhibition at the New Orleans Museum of Art, April-August 2003. This was a facade, 17 meters long, of Thomas Jefferson's home, Monticello, with two rear-projected, tracked, virtual-environment displays. It was experienced by 110,000 visitors. Please see <http://www.cs.unc.edu/~ibr/projects/NOMA/index.html> for details.

PATENTS

Fuchs, Henry, Anselmo A. Lastra, Jan-Michael Frahm, Nate Michael Dierk, David Paul Perra, "Low latency stabilization for head-worn displays," UA Patent number 9,898,866B2, issued February 20, 2018.

Popescu, Voicu Anselmo Lastra, John Eyles, "Methods and apparatus for rendering images using 3D warping techniques," US Patent number 6,756,993, issued June 29, 2004.

CONFERENCE (un-refereed)

Anselmo Lastra, Ascanio d'Andrea and Daniele Sepio, Swivel-Chair VR: A Portable HMD System (poster presentation), Virtual Systems and Multimedia (VSMM), Brisbane, Australia, September 2007.

Jerald, Jason Andrew Fuller, Anselmo Lastra, Mary Whitton, Luv Kohli, Fred Brooks, Latency Compensation by Horizontal Scanline Selection for Head-Mounted Displays, Proceedings of the SPIE conference on The Engineering Reality of Virtual Reality, January 2007.

Hensley, Justin, Thorsten Scheuermann, Montek Singh, Anselmo Lastra, Interactive Summed-Area Table Generation for Glossy Environmental Reflections, *ACM SIGGRAPH 2005 Sketches and Applications*, 2005.

Wei-Chao Chen, Lars Nyland, Anselmo Lastra, Henry Fuchs, Acquisition of Large-Scale Surface Light Fields, *ACM SIGGRAPH 2003 Sketches and Applications*, July 2003.

David K. McAllister, Benjamin P. Cloward, Anselmo A. Lastra and Wolfgang Heidrich, Spatial Bi-Directional Reflectance Distribution Functions, *ACM SIGGRAPH 2002 Sketches and Applications*, July 2002.

Kok-Lim Low, Greg Welch, Anselmo Lastra and Henry Fuchs, Life-Sized Projector-Based Dioramas: Spatially Real and Visually Virtual, *ACM SIGGRAPH 2001 Sketches and Applications*, August 2001.

Nyland, Lars, Anselmo Lastra, David McAllister, Voicu Popescu, and Chris McCue, Capturing and Rendering of Real-World Scenes, Videometric and Optical Methods for 3D Shape Measurement, Photonics West, San Jose, CA, January 2001.

Daniel Aliaga and Anselmo Lastra, "Virtual Backdrops", Technical Sketch, Visual Proceedings, ACM SIGGRAPH, August, 1997.

Lastra, Anselmo, Henry Fuchs, and John Poulton, Harnessing Parallelism for High-Performance Interactive Computer Graphics, *Proceedings of NSF Workshop on Experimental Systems*, June 1996.

Lastra, Anselmo and Steve Molnar, PixelFlow: The Continuing Saga, *Proc. 1995 Parallel Rendering Symposium*, ACM SIGGRAPH, ISBN: 0-89791-74-1, 79, September 1995.

OTHER

Nyland, Lars and Anselmo Lastra, Visualizing the Real World, IEEE Computer Graphics and Applications, Sept. 2001.

Nyland, Lars and Anselmo Lastra, Rangefinder / Digital Camera System Renders Real-World Scenes, Laser Focus World, June 2001.

Lastra, Anselmo A., Technology for Virtual Reality, *Siggraph 97 Course Notes*, A. Lastra, ed., ACM SIGGRAPH, August 1997.

Olano, Marc, Anselmo Lastra, and Jon Leech, Procedural Primitives in a High Performance, Hardware Accelerated, Z-Buffer Renderer, UNC Computer Science Technical Report TR97--040, 1997.

Lastra, Anselmo, Technology for Virtual Reality, *Siggraph 96 Course Notes*, A. Lastra, ed., ACM SIGGRAPH, August 1996.

Saxe, Edward, Merlin Hughes, and Anselmo Lastra, "Higher-Order Color Interpolation for Real-Time Radiosity Display", UNC-CH Department of Computer Science Technical Report TR96-023, 1996.

Lastra, Anselmo, Technology for Virtual Reality, *SIGGRAPH 95 Course Notes*, A. Lastra, ed., ACM Siggraph, August 1995.

Lastra, Anselmo, Technology for Virtual Reality, *Siggraph 94 Course Notes*, A. Lastra, ed., ACM Siggraph, 3.1 - 3.25, 1994.

Holloway, Richard and Anselmo Lastra, *Virtual Environments: A Survey of the Technology*, Eurographics Technical Notes, ISSN 1017-4656, 1-50, 1993.

Lastra, A. A., and Starmer, C. F., Architectural Support for Parallel Debugging, Department of Computer Science Technical Report CS-1989-10, Duke University, Durham, NC, 1989.

TEACHING (number of times taught)

Undergraduate level

Digital Computer Design at UNC (4) – designed and inaugurated new course
Systematic Programming (2) at UNC
First Year Seminar using LEGO Robots (2) at UNC
3D Computer Animation (1) at UNC
Software Design (1), at Duke University
Computer Graphics (2), at Duke University – designed and inaugurated new course

Graduate level

Programming Methodology (3), at Duke University
Advanced Image Generation (5), at UNC – advanced research preparation course
Graphics Hardware (2) at UNC
Computer Architecture (3) at UNC

GRANTS

Principal Investigator, Imperceptible and RealTime 3D Scanning for Biometric Identification and Verification, Department of Homeland Security, \$215,291, 2010-2011.

Principal Investigator, Power-Aware Graphics Hardware, National Science Foundation, \$350,000, 2007-2012.

Principal Investigator, Fifth Generation Graphics Architectures, National Science Foundation, \$350,000, 2003-2007.

Co-Principal Investigator, Tera-Pixels: Using High-resolution Pervasive Displays to Transform Collaboration and Teaching, National Science Foundation, \$1,066,316, 2003-2008.

Principal Investigator, ITR: Collaborative Research: Image-Based Rendering in Forensic Science, Education, and Historical Preservation, National Science Foundation, collaborative research with the University of Virginia, \$1,190,818 to UNC, \$1,439,054 total, 2002-2006.

Senior Investigator (and proposal editor), Real-Time Long-Distance Terascale Computation for Full Bandwidth Tele-Immersion, National Science Foundation, \$2,651,500, 2001-2004.

Co-Principal Investigator with Frederick P. Brooks Jr. and Dinesh Manocha – Real-Time Walkthroughs of Serious Synthetic Environments, National Science Foundation, \$470,000, 1999-2002.

Co-Principal Investigator with Dinesh Manocha, ASCI Program, Department of Energy, 1999-2002.

Co-Principal Investigator (and proposal editor) with Henry Fuchs, Frederick P. Brooks Jr., Gary Bishop and Dinesh Manocha - Acquisition of a Graphics Supercomputer for Synthetic Environments Serving Science and Engineering, National Science Foundation, \$1.3 million, 1998-2001.

Co-Principal Investigator (and proposal editor) – Intel equipment grant, \$2.5 million, 1997-2000.

Senior Investigator – ImageFlow: Real-Time Image-Based Rendering, DARPA, \$2.1 million, 1995-1998.
Senior Investigator – ImageFlow: Real-Time Image-Based Rendering, National Science Foundation, \$2.1 million, 1995-1999.
Principal Investigator - Computer Solutions to Multipoint Linkage Analysis, National Institutes of Health grant 2P50AGO5128, 1990 - 1992.
Principal Investigator - Cray Research Grant, 1991.

PROFESSIONAL ACTIVITIES

Steering committee for the BeingThere Center, 2010-
Steering committee for High Performance Graphics, 2009-
Co-Treasurer, High Performance Graphics 2012, Paris
Associate Editor and Tutorials Editor, IEEE Computer Graphics and Applications 2007-2011
Associate Editor, IEEE Transactions on Visualization and Computer Graphics, 2003-2007
Co-Treasurer, High Performance Graphics 2011, Vancouver
Steering committee for Graphics Hardware conferences, 2005-2009
Co-Treasurer, High Performance Graphics 2010, Saarbrücken
Program committee, Computer Applications in Archaeology (CAA) 2010
Program committee, Symposium on Interactive 3D Graphics (I3D) 2010
Co-Treasurer, High Performance Graphics 2009, New Orleans
Program Committee, eHeritage 2009
Program Committee, Symposium on Interactive 3D Graphics (I3D) 2009
Co-Treasurer, Graphics Hardware 2008, Sarajevo
Program Committee, Symposium on Interactive 3D Graphics (I3D) 2008
Co-Treasurer, Graphics Hardware 2007, San Diego
Program Committee, Symposium on Interactive 3D Graphics (I3D) 2007
Program committee, SIAGC 2006
Program committee, 3DPVT 2006
Program committee, ACM SIGGRAPH Video Game Symposium 2006
Program committee, Symposium on Point-Based Graphics 2006
Program committee, SIBGRAPI 2006
Program committee, EVA 2006, Austria
Program committee, Graphics Hardware 2006
Program committee, Symposium on Interactive 3D Graphics (I3D) 2006
Program committee, GRAPP 2006
Program committee, Symposium on Point-Based Graphics 2005
Program committee, Graphics Hardware 2005
Co-Chair, Symposium on Interactive 3D Graphics (I3D) 2005, Washington, DC
Program committee, 3DIM 2005
Co-Chair, Graphics Hardware 2004, Grenoble, France

Co-Chair, Workshop on General Purpose Computation on Graphics Processors, 2004, Los Angeles, CA
Program committee, Symposium on Point-Based Graphics 2004
Chair, Technical Outreach, SIGGRAPH (2003-2005)
Chair, Graphics Hardware 2002, Saarbrücken, Germany
Program co-Chair, Graphics Hardware 2001, Los Angeles
Courses committee, ACM SIGGRAPH 2001
Chair, courses, ACM SIGGRAPH 2000
Papers co-Chair, SIGGRAPH/Eurographics Workshop on Graphics Hardware 2000
Program committee, Immersive Projection Technology Workshop, 2000
Courses committee, ACM SIGGRAPH 99
Courses committee, ACM SIGGRAPH 98

INVITED TALKS

Keynote, *Title Bringing the Real to Virtual Reality*, SIACG 2006.
Capstone, *Modeling the Real World*, SIBGRAPI 2002.
Invited speaker, *Image-Based Rendering and Virtual Environments at UNC Chapel Hill*, EDGE Conference, IBM T.J. Watson Research Labs, April 2002.
Keynote, *Bringing the Real to Virtual Reality*, International Conference for Artificial Reality and Telepresence, Taipei, Taiwan, October 2000.
Image-Based Rendering, Beijing Simulation Center, Beijing, China, May 1999.
Invited speaker, *All the Triangles in the World*, Cornell Workshop on Rendering Perception and Measurement, Ithaca, NY, April 1999.
Capstone, *Virtual Reality: What does the Future Hold?*, Cyberconf 96, Madrid, June 96.
Invited speaker, *PixelFlow: The Continuing Saga*, Parallel Rendering Symposium, September 1995.

PAST GRADUATE STUDENTS (including MS theses)

Daniel Aliaga, Ph.D., UNC, 1998
Greg Coombe, PhD, 2007
Paul Keller, M.S., UNC, 1996
Mark Harris, PhD, UNC 2003
Justin Hensley, PhD, UNC 2007
Karl Hillesland, PhD, UNC, 2005
John Houde, M.S., Duke, 1990
Kok-Lim Low, Ph.D., UNC, 2006
David K. McAllister, Ph.D., UNC, 2002
Jonathan McAllister, M.S., UNC, 1996
Carl Mueller, Ph.D., UNC, 2000

Sandra Dawson Miller, M.S., Duke, 1990
Thomas Marcolin Olano, Ph.D., UNC, 1997
Jeff Pool, Ph.D., UNC, 2012
Voicu Popescu, Ph.D., UNC, 2001
Joshua Steinhurst, PhD, 2007
Meredith L. Smith, M.S., Duke, 1991

UNIVERSITY SERVICE

Faculty Executive Committee
Committee on Faculty Research and Study Leaves
Faculty Council
SACS Accreditation Committee, UNC Graduate School

DEPARTMENTAL SERVICE

Chairman
Director of Graduate Studies
Graduate Studies Committee
Graduate Curriculum and Planning Committee
Conflicts of Interest Committee
Undergraduate Curriculum Committee
Department Facilities Committee
Graduate Admissions Committee (twice)

PUBLIC SERVICE

2003 – Exhibition at New Orleans Museum of Art to celebrate the 200th anniversary of the Lewis and Clark expedition. Please see item under exhibitions in the publications section.

2004 – Documented a pre-Roman Faliscan tomb in central Italy by creating a 3D geometric model. This work is in collaboration with Professor Nicola Terrenato of the Department of Classics.

2005-6 – Technical advisor on a team to document the Suburban Baths on the Herculaneum archaeological site by creating a 3D model. This work is in collaboration with the British School at Rome.

TESTIMONY AND EXPERT REPORTS

- 2010-11 S3G v. Apple in the International Trade Commission (Apple), Inv. No. 337-TA-724, filed May 28, 2010. Was deposed.
- 2012 GPH v. Sony in the International Trade Commission (Sony), Inv. No. 337-TA-836, filed March 12, 2012.

- 2013-14 GPH v. Toshiba in the International Trade Commission (Toshiba, LG, Vizio), ITC Inv. No. 337-884, filed May 17, 2013. Was deposed and testified at hearing.
- 2013-14 Micrografx v. Google in the Federal Court of the Northern District of Texas (Google), case 3:13-cv-03595-N. Was deposed for Inter Partes Review.
- 2014-15 Ziilabs v. Samsung in the Eastern District of Texas (Samsung), case 2:14-cv-00203-JRG-RSP, filed March 10, 2014. Was deposed.
- 2014-17 Advanced Micro Devices v. LG Electronics in the Northern District of California (LG), case 3:14-cv-1012-SI, filed December 28, 2015.
- 2015 ART+COM vs. Google. Inter Partes Review of RE44,500 (Google).
- 2016 Advanced Silicon Technologies v. Texas Instruments and Renesas in the International Trade Commission (TI and Renesas), filed December 28, 2015. Inv. No. 337-TA-984.
- 2017 Advanced Micro Devices v. LG Electronics, MediaTek, Sigma Devices, and Vizio in the International Trade Commission (LG, MediaTek, Sigma, and Vizio). Inv. No. 337-TA-1044, filed January 24, 2017. Was deposed and testified at hearing.
- 2018 Ziilabs v. NVIDIA in the International Trade Commission (NVIDIA). Inv. No. 337-TA-1099, filed December 29, 2017.
- 2020 Infernal Technology vs. Sony Interactive Entertainment in the Eastern District of Texas (Sony), case 2:19-cv-248-JRG, filed July 11th, 2019. Was deposed and testified at trial.
- 2020 GPU++ v. Qualcomm. Inter Partes Review of 8,988,441 (Qualcomm).
- 2020 Kewazinga v. Google in the Southern District of New York (Google), case 1:20-cv-01106-LGS, filed February 7th, 2020. Was deposed.
- 2022 AMD vs. Realtek in the International Trade Commission, Investigation No. 337-TA-1318.

COMPANIES FOR WHICH I HAVE CONSULTED

- AMD
- Apple
- Google
- LG
- MediaTek
- NVIDIA
- Renesas
- Samsung
- Sigma Devices
- Sony
- Qualcomm
- Texas Instruments
- Toshiba
- Vizio