

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

**PATENT OWNER'S PRELIMINARY RESPONSE
PURSUANT TO 37 C.F.R. § 42.107(A)**

TABLE OF CONTENTS

	Page(s)
I. INTRODUCTION	1
II. GRAPHICS PROCESSING BACKGROUND.....	3
A. Basic Graphics Processing	3
B. Texture Mapping and Texels.....	4
C. Texture Compression	5
1. Images and Blocks	6
2. Graphics Versus General-Purpose Processing.....	7
III. THE '628 PATENT.....	8
A. Drawbacks in Prior Art Texture Compression Methods.....	8
B. The '628 Patent's Disclosed Solutions	10
1. Disjoint Subsets.....	10
2. Index Values	13
C. The Challenged '628 Patent Claims.....	16
IV. THE PETITION.....	17
A. Sorgard	17
B. FXT1.....	21
V. PETITIONER HAS FAILED TO ESTABLISH THE REQUIRED REASONABLE LIKELIHOOD OF SUCCESS FOR ANY GROUNDS....	22
A. Ground 1: Sorgard and the Knowledge of a POSITA Does Not Disclose the Claimed First and Second Index Values	22
1. Sorgard's Fading Pattern Index is Not Either of the Claimed Index Values	23
2. Petitioner Fails to Identify a Second Index Value	26
B. Grounds 2 & 4: FXT1 Does Not Disclose the Claimed Disjoint Subsets.....	28
1. FXT1 Does Not Disclose the Claimed Subsets	29

a.	FXT1 Does Not Disclose The Claimed Blocks or Subsets Under Correct Construction of “Texture Block” and “Subset”	29
b.	Treating FXT1’s “Image” as the Claimed “Texture Block” Would Render the Resulting System Inoperable	34
2.	Petitioner’s Obviousness Argument is Pure Hindsight Without Any Motivation	36
VI.	CONCLUSION.....	38

TABLE OF AUTHORITIES

	Page(s)
Federal Cases	
<i>Aventis Pharma S.A. v. Hospira, Inc.</i> , 675 F.3d 1324 (Fed. Cir. 2012)	30
<i>Hill-Rom Services, Inc. v. Stryker Corporation</i> , 755 F.3d 1367 (Fed. Cir. 2014)	30
<i>InTouch Techs., Inc. v. VGo Communs., Inc.</i> , 751 F.3d 1327 (Fed. Cir. 2014)	37
<i>KSR International Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	36
<i>Merck & Co. v. Teva Pharms. USA, Inc.</i> , 395 F.3d 1364 (Fed. Cir. 2005)	27
<i>SimpleAir, Inc. v. Sony Ericsson Mobile Communs. AB</i> , 820 F.3d 419 (Fed. Cir. 2016)	27
<i>Starhome GmbH v. AT&T Mobility LLC</i> , 743 F.3d 849 (Fed. Cir. 2014)	30

TABLE OF EXHIBITS

Exhibit	Description
2001	Declaration of Dr. Anselmo Lastra and CV
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.
2003	Edwin Earl Catmull. 1974. “A Subdivision Algorithm For Computer Display of Curved Surfaces.” Ph.D. Dissertation. The University of Utah
2004	Ziyad S. Hakura and Anoop Gupta. 1997. The design and analysis of a cache architecture for texture mapping. <i>SIGARCH Comput. Archit. News</i> 25, 2 (May 1997), 108–120.
2005	Michael Doggett. 2012 “Texture Caches” In: <i>IEEE Micro</i> . Vol. 32, No. 3. pp. 136-141.
2006	fsdeveloper.com wiki (https://www.fsdeveloper.com/wiki/index.php/Texture_formats_overview)
2007	OpenGL Wiki (www.khronos.org/openg/wiki/S3_Texture_Compression)
2008	Rui Wang Wei Hua, et al. 2012. Compressing repeated content within large-scale remote sensing images. <i>Vis. Comput.</i> 28, 6–8 (June 2012), 755–764.

I. INTRODUCTION

The Board should deny Realtek’s Petition because it fails to establish the required reasonable likelihood of success under any of the asserted grounds.

Ground 1: All of the challenged claims require a “first index value” and a “second index value,” each of which is used to interpolate between different sets of color endpoints to determine the color of an individual “texel.” Ground 1 relies solely on Sorgard’s fading pattern index as teaching each of these index values. But Sorgard does not disclose the claimed index values (and Ground 1 therefore must fail) for at least two reasons.

First, Sorgard’s fading pattern cannot be either of the claimed index values as Petitioner asserts because it is not used to interpolate the colors of texels between endpoints as required by the claims. Instead, Sorgard interpolates based on a fixed formula, not based on any individual index value (or other parameter).

Second, regardless of whether Sorgard’s index value can be a claimed index value, each challenged claim requires two different index values: “a *first* index value for the first texel” and “a *second* index value for the second texel.” But the Petitioner identifies and relies on the *same single* Sorgard fading pattern as somehow disclosing the claimed *two different* index values. For this reason alone, Ground 1 fails.

Grounds 2 and 4: In addition to the foregoing two index values, each challenged claim also requires that a texture image be divided into texture blocks, and that a texture block be further subdivided into two or more disjoint subsets. Each of the remaining Grounds 2 and 4 (the Petition does not include a Ground 3) relies solely on FXT1 as teaching the claimed texture block and disjoint subsets, but it cannot because FXT1 does not disclose any subsets.

Rather, FXT1—by its own words—merely discloses dividing an “image” into “texture blocks.” FXTI does not disclose or suggest further dividing those texture blocks into subsets as required by all challenged claims. Indeed, the ’628 Patent specifically discusses and distinguishes FXT1 (and other block-based approaches) as dividing a texture image into texture blocks but not further dividing those texture blocks into smaller subsets.

To overcome this fatal flaw, Petitioner relabels FXT1’s disclosed complete “image” as a texture block, and relabels FXT1’s disclosed “blocks” as subsets. But Petitioner offers no claim construction to support redefining “block” and “subset” in this manner, and there is none. Both the well-understood ordinary meanings of image, block, and subset, as well as the manner in which those terms are used in the ’628 Patent, are contrary to Petitioner’s attempt to find subsets in FXT1. Nor does Petitioner provide any rationale for merely ignoring the claims’ requirement of a second-level subdivision. For these reasons, Grounds 2 and 4 also fail.

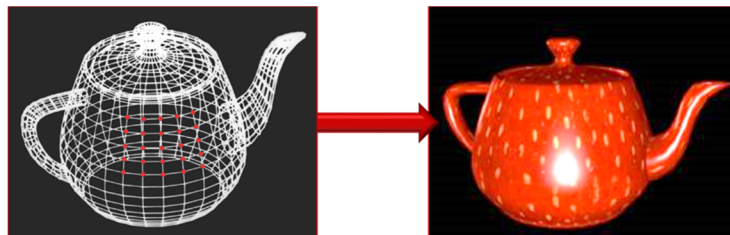
Accordingly, the Board should deny institution of this proceeding.

II. GRAPHICS PROCESSING BACKGROUND

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. Basic Graphics Processing

Graphics processing involves manipulating and rendering 3D graphical objects onto a 2D display screen. Ex. 2001, ¶ 38. 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. 2001, ¶ 38. 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. 2001, ¶ 38.

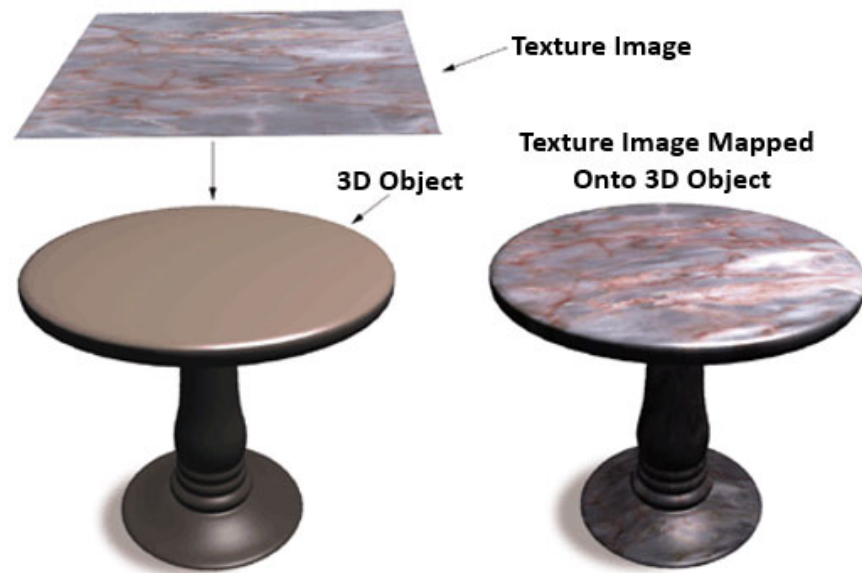


The conversion of a 3D shape consisting of a mesh of triangles into a photorealistic 2D image consisting of an array of pixels happens in a number of

processing stages. Ex. 2001, ¶ 39. 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. 2001, ¶ 39.

B. Texture Mapping and Texels

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; Ex. 2001, ¶ 40. Texture mapping is the process of applying a texture image to a group of pixels representing a surface to be displayed. Ex. 1001 at 3:61-4:2; Ex. 2001, ¶ 40. Texture mapping reproduces the detail by mapping a texture image onto these pixels. Ex. 1001 at 3:61-4:2; Ex. 2001, ¶ 40. 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. Ex. 2001, ¶ 40.



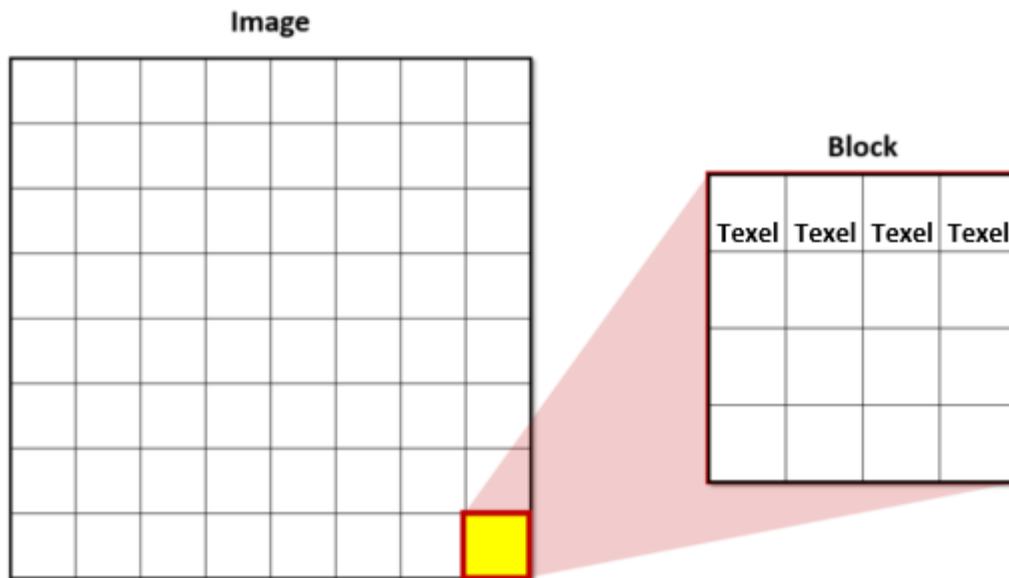
The texture image is made up of “texels.” Ex. 1001 at 3:48-53; Ex. 2001, ¶ 41. Texels are similar to pixels: the same way a “pixel” is an element of a picture, a texel is an element of a texture image. Ex. 1001 at 3:48-53. The texel represents the color or other surface characteristic of a corresponding pixel on the surface to be displayed. *Id.* at 3:65-4:2.

C. Texture Compression

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; Ex. 2001, ¶ 42.

1. Images and Blocks

In one prior art approach, for the GPU to be able to load texels as needed, the texture image is divided into smaller fixed size blocks (*e.g.*, 4 texels x 4 texels or 4 texels x 8 texels, 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; Ex. 2001, ¶ 43.



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 and presents difficulties with random access of blocks. *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.”); Ex. 2001, ¶ 44.

2. Graphics Versus General-Purpose Processing

Graphics processing differs from general-purpose processing for several reasons. Ex. 2001, ¶ 45. 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.*, 2048x2048 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.”). Ex. 2001, ¶ 45.

This was especially true in 2006, the year of priority of the ’628 patent, when processing power was much more limited than it is today. Ex. 2001, ¶ 46. 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 can be accessed without consuming too much memory, and certainly operating on the entire image at once is not possible, and especially was not in 2006. *See* Ex. 2001, ¶¶ 46-47; Ex. 1003, ¶ 30.

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); Ex. 2001, ¶ 48. Decompressing a full texture image when only a small component is needed would be wasteful and is not done. Ex. 2001, ¶ 48.

III. THE '628 PATENT

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 texel colors in the respective subsets. Ex. 2001, ¶¶ 49-50.

A. Drawbacks in Prior Art Texture Compression Methods

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; Ex. 2001, ¶ 51.

For example, among the prior art approaches discussed and distinguished are block-based techniques such as DXTC (also known as S3TC) and FXT1 (the same FXT1 approach that forms the basis for Petitioner’s Grounds 2 and 4). Ex. 1001 at 2:24-52; Ex. 2001, ¶ 52. As the ’628 Patent discloses, DXTC “is a block-based texture compression scheme” that divides a texture image into 4x4 pixel blocks. Ex. 1001 at 2:24-32. FXT1 is another block-based texture compression scheme that “essentially extends DXTC with some additional block types.” *Id.* at 2:46-50. As with DXTC, there is only a single level of division (*e.g.*, from the texture image to blocks). Ex. 2001, ¶ 53.

The DXTC and FXT1 block-based approaches suffered from numerous drawbacks. Ex. 2001, ¶ 54. For example, DXTC has difficulty with textures that have many different color hues within one block. *Id.* at 2:34-35. DXTC also has difficulty also has problems handling blocks that have multiple separate color gradients at different orientations, as accurate compression of one gradient must usually be sacrificed. *Id.* at 2:39-44. Additionally, the low precision of DXTC’s color endpoints and its small number of interpolants can produce noise on gradients. *Id.* at 2:35-38.

The FXT1 block-based scheme is an extension of DXTC but did not provide any image quality gains over DXTC. Ex. 2001, ¶ 55. As a result, it received limited industry support. Ex. 1001 at 2:46-52.

B. The '628 Patent's Disclosed Solutions

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 individual texels within a subset, each individual index indicating how to interpolate and calculate the color for its corresponding texel. Ex. 1001 at 6:5-10, 10:61-11:11; Ex. 2001, ¶ 56.

1. Disjoint Subsets

The '628 Patent describes that while prior art methods, such as DXTC and FXT1, compressed images into blocks of texels (*e.g.*, 4x4 or 8x8 blocks), the '628 Patent discloses inventive embodiments that further subdivide the blocks into “subsets” of varying shapes and sizes. Ex. 2001, ¶ 57; Ex 1001 at 6:5-15, 6:27-42.

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; Ex. 2001, ¶ 58.

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 [texels], (e.g., 4x4 [texel] of 8x8 [texel] 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.

Ex. 1001 at 5:66-6:9; *see also* Ex. 2001, ¶ 59.

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. Ex. 2001, ¶ 60; Ex. 1001 at Fig. 8. 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

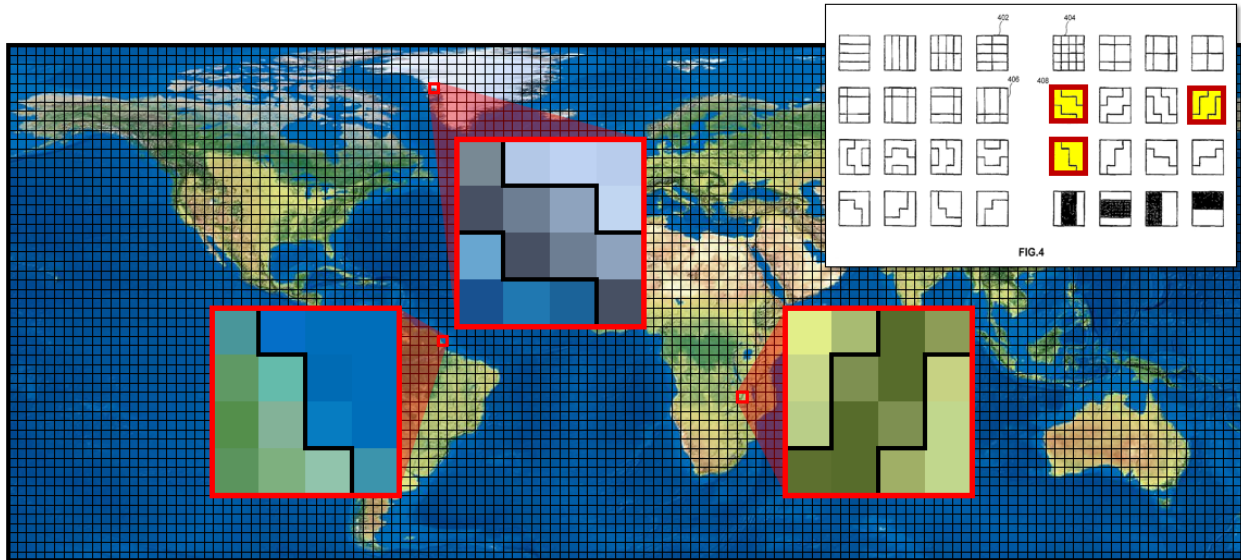
FIG. 8

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

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. Ex. 2001, ¶ 61. Furthermore, it provides improvement in the ability to compress less common types of texture data (such as non-photorealistic textures, cartoons, etc.).” Ex. 1001 at 6:10-16.

Below is an example of the use of the disjoint subsets of the ’628 Patent. Ex. 2001, ¶ 62. The ability to match the color contours of an image using disjoint subsets is well illustrated by the block of texels on the coast of Greenland. *Id.* The

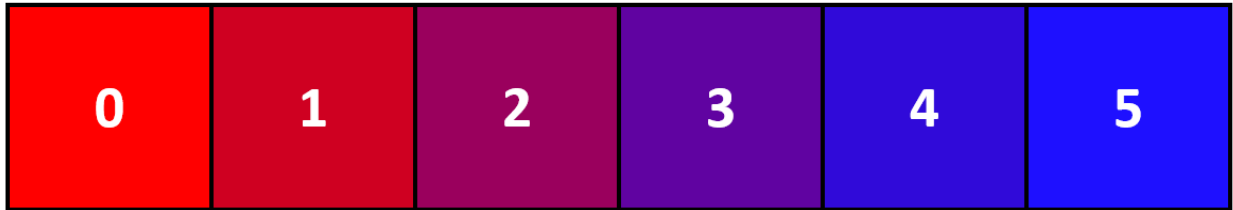
three partitions match the three different groups of colors (blue, grey, and near white), thus reducing the error. *Id.*



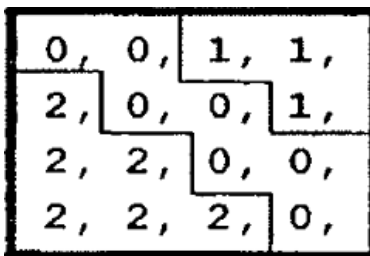
2. Index Values

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; Ex. 2001, ¶ 63. 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.*, Ex. 1001 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. Ex. 2001, ¶ 63.



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. Ex. 1001 at 11:50-54; Ex. 2001, ¶ 64.

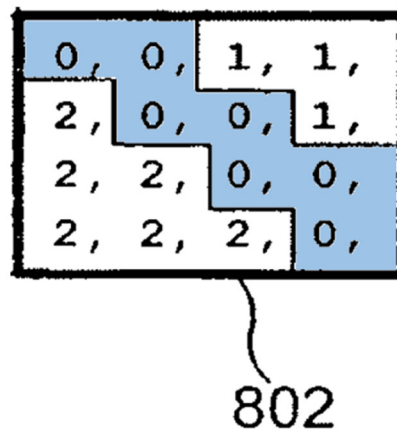


802

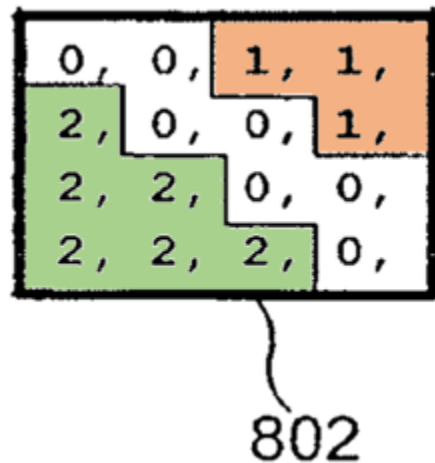
Ex. 1001 at 10:32-36; 10:53-60; 9:40-57; Fig. 8k.

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. Ex. 2001, ¶ 65. 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.*



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.*; Ex. 2001, ¶ 66.



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; Ex. 2001, ¶ 67.

C. The Challenged '628 Patent Claims

The Petition challenges independent claim 7 and its dependent claims 8-12. Consistent with the foregoing, independent claim 7 and its dependent claims require, further dividing a block into different subsets and using different index values to interpolate and determine the color of texels in different subsets:

A computer processing system comprising:

a processor and a memory, wherein the processor is configured to:

receive a compressed texture block, wherein *the compressed texture block includes two or more disjoint subsets* into which data in the compressed texture block is to be unpacked, wherein the two or more disjoint subsets include a first subset having a first set of color

endpoints and a second subset having a second set of color endpoints;
and

individually decompress each of the two or more disjoint subsets to
form texels, wherein the decompressing includes:

*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 at Claim 7.

IV. THE PETITION

Realtek's Petition asserts the following Grounds 1, 2, and 4:

Ground	Claims	Proposed Statutory Rejections
1	7-12	Obvious under 35 U.S.C. § 103 (a) over Sorgard in view of the knowledge of a person of ordinary skill in the art
2	7-12	Obvious under 35 U.S.C. § 103 (a) over FXT1 in view of the knowledge of a person of ordinary skill in the art
4	12	Obvious under 35 U.S.C. § 103 (a) over FXT1 in view of Reshetov and the knowledge of a person of ordinary skill in the art

Ground 1 relies primarily on the Sorgard reference, while Grounds 2 and 4 rely primarily on the FXT1 reference.

A. Sorgard

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; Ex.

2001, ¶¶ 68-69. Each block is further divided into two sub-blocks of 4x4 texels as shown below. Ex. 1004 at 24:23-38, Fig. 1.

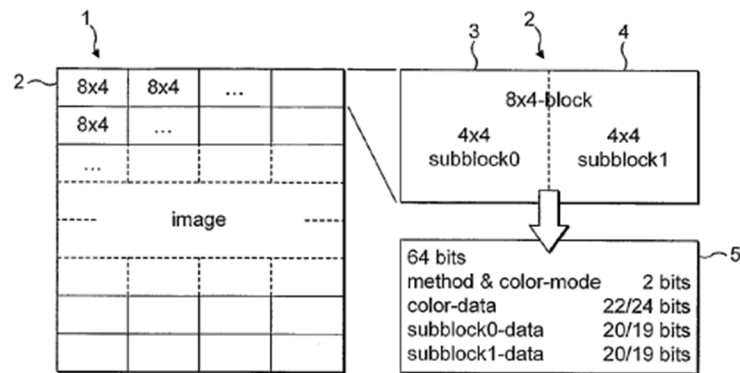
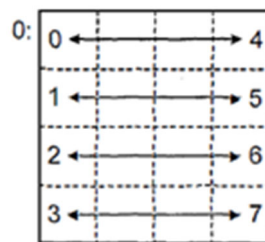


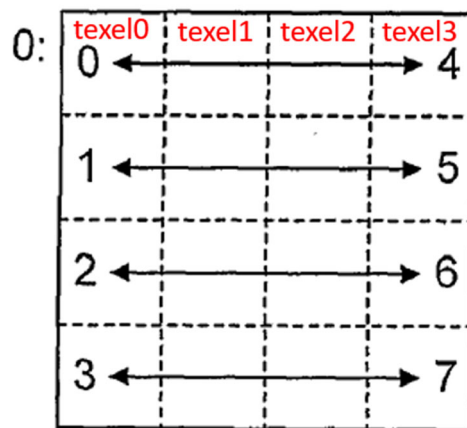
FIG. 1

Sorgard describes a compression and decompression scheme in which one of eight “fading patterns” is used to represent and calculate the colors of texels in a 4x4 block. Ex. 2001, ¶ 70. Sorgard’s fading pattern “comprises a set of four interpolation paths that each individually cover four texture elements.” Ex. 1004 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; *see also* Ex. 2001, ¶ 70. The Petition asserts that this 4x4 block satisfies the claimed “block” and that each interpolation path is a “subset” of that block.

Sorgard labels the four texels within each interpolation path (from left to right) as texel0, texel1, texel2, and texel3. Ex. 2001, ¶ 71. The texels at the endpoints of the path are texel0 and texel3 (as annotated below using path 0-4). *Id.* The interior texels (the ones between the endpoints) are texel1, and texel 2. *Id.*



Ex. 1003 at Fig. 5 (annotated).

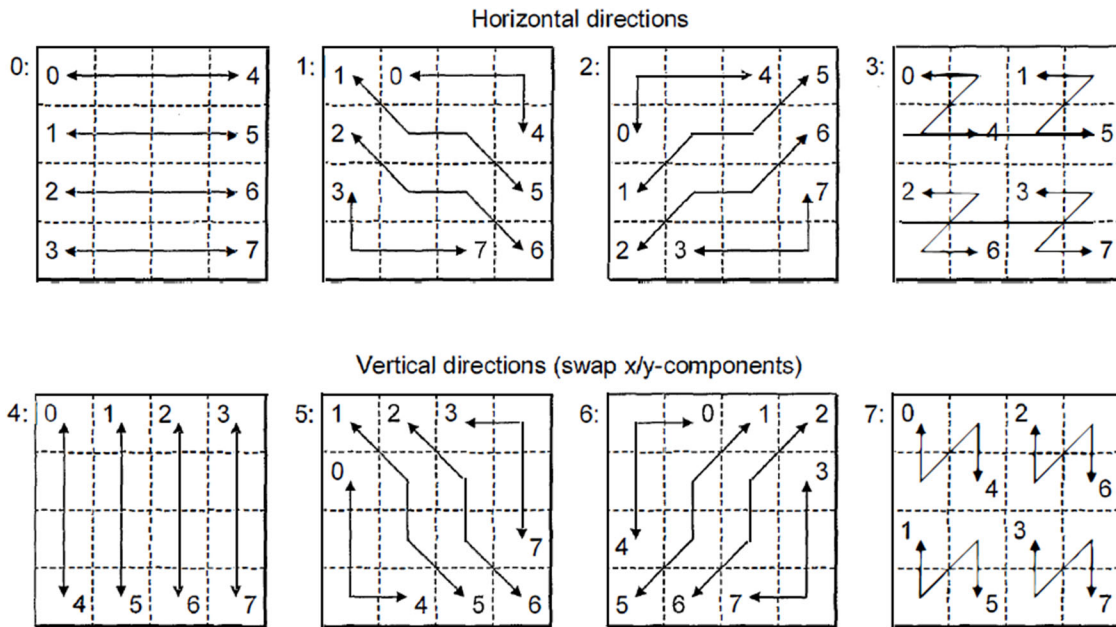
To determine the colors of each texel in a path during decompression, Sorgard first describes that there is a color associated with each endpoint of a path, with color[x] assigned to texel0 and color[y] assigned to texel3. Ex. 2001, ¶ 72. Then, to determine the colors of the interior texels, a preset formula is used: the color of interior texel1 is always $\frac{5}{8}$ of color[x] plus $\frac{3}{8}$ of color[y], while the

color of interior texel2 is always 3/8 of color color[x] plus 5/8 of color[y]. Sorgard sets forth these color formulas as follows:

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

Ex. 1003 at 34:1-8.

Sorgard also describes that there are eight possible fading patterns that may be used to compress and decompress the color data for the 4x4 block. Ex. 2001, ¶ 73. These different fading patterns are labeled as number 0 through 7 and are illustrated below:



Ex. 1003 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.

According to the Petition, the number of the selected fading pattern (any of 0 through 7) satisfies both the claimed first index value and the claimed second index value. For example, if fading pattern 3 was used to decompress the 4x4 block, under the Petition's Ground 1, each path within that block is a disjoint subset, and the number "3" is both the first index value and the second index value.

B. FXT1

"FXT1 Texture Compression Technology White Paper" ("FXT1") is a paper describing at a very high level a block-based texture compression scheme. Ex. 2001, ¶¶ 74-75. As discussed, 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.

FXT1 explicitly describes dividing an entire image into blocks: "FXT1™ texture compression works by dividing an image into multiple 4x4 and/or 4x8 texel blocks." Ex. 1005 at 7; Ex. 2001, ¶¶ 75-77. To overcome this deficiency, the Petition's Grounds 2 and 4 assert that what FXT1 explicitly calls an "image"

should instead be considered a “texture block,” and that what FXT1 explicitly calls a “texture block” should instead be considered a “subset” of that texture block.

V. PETITIONER HAS FAILED TO ESTABLISH THE REQUIRED REASONABLE LIKELIHOOD OF SUCCESS FOR ANY GROUNDS

A. Ground 1: Sorgard and the Knowledge of a POSITA Does Not Disclose the Claimed First and Second Index Values

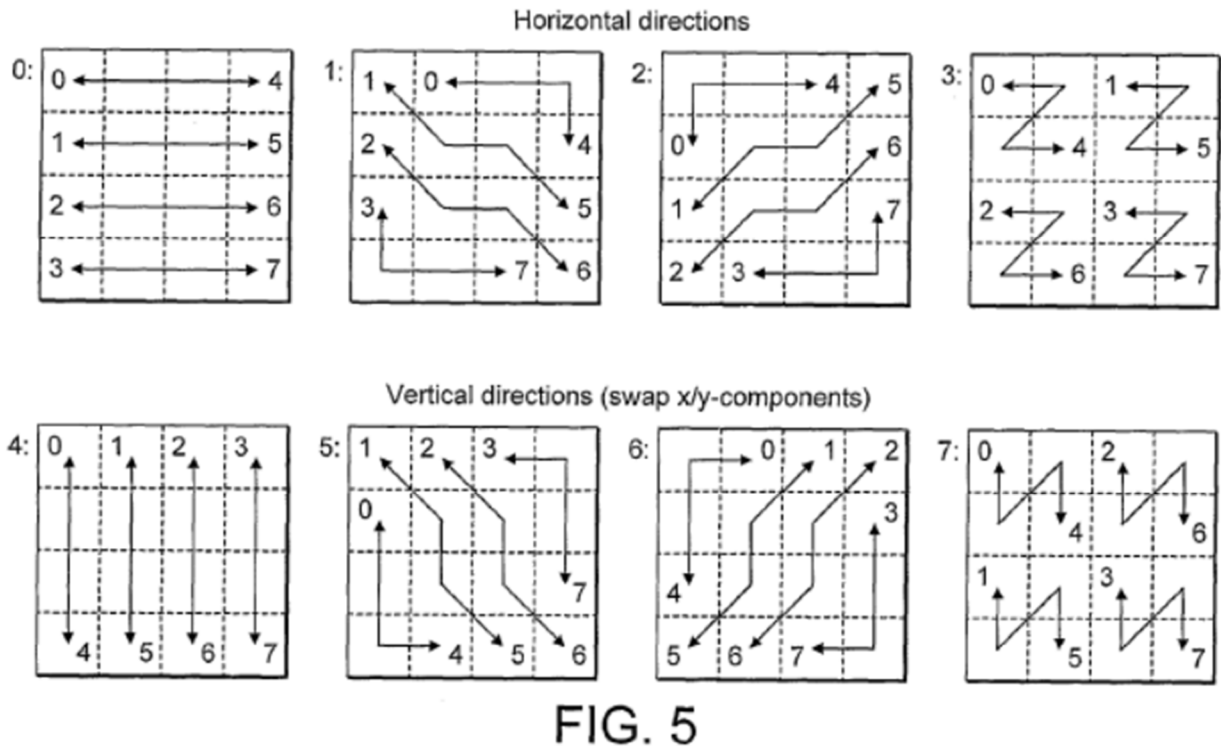
Petitioner asserts that Sorgard teaches the claimed first and second index values, but Sorgard does not for at least the following two reasons. First, Sorgard does not disclose using an “index” value to indicate how to interpolate the color of a texel between the endpoints. Rather, Sorgard uses a fixed linear equation, meaning that the amount of interpolation is predetermined and constant for any given texel between the color endpoints. Further, the indexes pointed to by Petitioner are not indexes for texels – they are indexes for fading patterns or interpolation direction.

Second, all challenged claims require that the first and second texels are interpolated using two different index values, “a first index value for the first texel,” and “a second index value for the second texel.” Ex. 1001, Claim 7. However, Petitioner points to the same index value as the purported first and second index values.

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

The purported index value identified by Petitioner cannot be the claimed “first index value” *or* the claimed “second index value” because it is not used in the way required in the claims. The claims require “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 similarly require this for the second index value and second texel. Ex. 1001, Claim 7 (emphasis added). Sorgard on the other hand uses an interpolation formula to determine color, but Sorgard's interpolation formula does not depend on the alleged index value. Ex. 2001, ¶¶ 79-80. Rather, the formula is fixed regardless of index value and thus Sorgard does not “interpolat[e] between color endpoints . . . based on [an] . . . index value.”

Petitioner alleges that “a 3-bit index is used to select one of the eight fading patterns in Figure 5”:



Ex. 1004, Fig. 5; Pet. at 36. Petitioner then argues that “[f]or each texel, this index determines the interpolation formula used to derive its color.” Pet. at 36. That is incorrect. Ex. 2001, ¶¶ 81-84. These indexes are only used to determine the colors of the endpoints of the 4-texel paths but have no impact on the interpolation formula used for the intermediate texels: “Colour-indexes for the endpoints are then stored in a 2-bit per element array called indexes.” Ex. 1004 at 33:55-56; *see also* Ex. 2001, ¶¶ 79-80. The question relevant to the claims’ validity, then, is what method is used by Sorgard to determine the colors of the intermediate texels, *i.e.*, the claimed first and second texels, whose colors are “between color endpoints” in the language of claim 7, and does that method satisfy the claim

requirement of doing so “by interpolating between color endpoints . . . based on a first index value” (and “second index value”).

The answer is “no.” Ex. 2001, ¶¶ 79-80. The colors of the two remaining texels as disclosed by Sorgard are determined by an interpolation formula that is always the same, and not dependent on any index: “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); Ex. 2001, ¶¶ 79-80. As seen above, the intermediate texel colors are determined “by interpolating between color endpoints” in a way that is *not* “based on a first index value” or “based on second index value.” *Id.* Instead, it is determined by interpolating between color

endpoints exclusively based on a fixed formula that blends color endpoints at a $3/8$ and $5/8$ ratio. Ex. 2001, ¶¶ 80, 82-83.

For all indexes, all fading patterns, and all endpoint colors, the $3/8$ and $5/8$ blend is the sole, fixed, interpolation formula for the intermediate (*i.e.*, the claimed first and second) texels, and so there is no disclosure of interpolating the intermediate texels based on an index. Ex. 2001, ¶¶ 80, 82-83. Consequently, the Board should decline to institute Realtek’s Petition with respect to Ground 1.

2. Petitioner Fails to Identify a Second Index Value

Petitioner identifies Sorgard’s 4x4 subblocks (e.g., subblock0 or subblock1 as the claimed “compressed texture block.” Pet. at 27 (annotated figure), 28 (asserting that each 4x4 texture block is divided into the claimed “disjoint subsets”); Ex. 2001, ¶ 86. Therefore, for Sorgard to invalidate, each 4x4 subblock must have at least two disjoint subsets, where each subset has a first index value for a first texel and a second index value for a second texel. *See* Ex. 1001, Claim 7; Ex. 2001, ¶ 86.

Additionally, the first and second texels cannot use the same index value. A POSITA would understand 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. 2001, ¶ 87. Claim interpretation

also requires that the “first index value” and the “second index value” be different index values, each specific to their respective texel. *See SimpleAir, Inc. v. Sony Ericsson Mobile Communs. AB*, 820 F.3d 419, 431 (Fed. Cir. 2016) (citing *Bd. of Regents of the Univ. of Tex. Sys. v. BENQ Am. Corp.*, 533 F.3d 1362, 1371 (Fed.Cir. 2008) (“Different claim terms are presumed to have different meanings.”)). Additionally, allowing the same index value to read on both the “first index value” and “second index value” would render the “second index value” superfluous. *Merck & Co. v. Teva Pharms. USA, Inc.*, 395 F.3d 1364, 1372 (Fed. Cir. 2005) (“A claim construction that gives meaning to all the terms of the claim is preferred over one that does not do so.”).

But Petitioner only points to one alleged index value, and relies on it for both the alleged first and second texels: “Sorgard uses ***an index*** to determine the fading pattern of the interpolation paths. Ex. 1004 at 33:43-47. For example, a 3-bit index is used to select one of the eight fading patterns in Figure 5. Ex. 1003, ¶ 99. ***For each texel, this index determines*** the interpolation formula used to derive its color.” Pet. at 36 (emphasis added); *see also* Pet. at 35, 38 (for “second index value for second the second texel,” merely referring up to the discussion for “first index value for the first texel”); Ex. 2001, ¶ 85. 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. Although this 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, which is Petitioner’s identified block. *Id.*; *id.* at 33:1-22; Ex. 2001, ¶¶ 88-89. Therefore, Sorgard does not disclose using two different index values, one respective to a first texel and a second respective to a second texel.

B. Grounds 2 & 4: FXT1 Does Not Disclose the Claimed Disjoint Subsets

FXT1 merely discloses dividing an original texture image into texture blocks, it does not disclose further dividing those texture blocks into disjoint subsets as required by all challenged claims:

FXT1TM texture compression works by *dividing an image into multiple 4x4 and/or 4x8 texel blocks*.

Ex. 1005 at 7 (emphasis added).

The Petition takes two approaches to overcome this fatal deficiency.

Primarily, the Petition tacitly construes the claimed “texture block” to include an original texture image and tacitly construes the claimed “disjoint subsets” to include “texture blocks.” But the Petition identifies no basis for rewriting these claim terms in this manner and there is none because Petitioner’s position is contrary to both the ordinary meaning and the ’628 Patent’s use of image, texture block, and subset.

As a backup argument, Petitioner also asserts that it would have been obvious to further divide FXT1’s texture blocks into subsets. But this assertion is based purely on impermissible hindsight, and the Petition provides no basis for its obviousness contention other than Petitioner’s own desire to invalidate the challenged claims.

1. FXT1 Does Not Disclose the Claimed Subsets

a. *FXT1 Does Not Disclose The Claimed Blocks or Subsets Under Correct Construction of “Texture Block” and “Subset”*

Contrary to the unstated underlying theory on which the Petition’s Ground 2 and 4 hinge, the correct construction of “texture block” does not include an original texture “image,” and the correct construction of “disjoint subsets” does not include blocks. Indeed, despite its burden of proof, the Petitioner does not even provide any claim constructions or arguments on these key issues. Nor could it

reasonably do so because under both their undisputed ordinary meaning and their use in the '628 Patent: (1) a “texture block” is not an original image, it is a division of an image; and (2) a “subset” is not a block, it is a division of a block. Ex. 2001, ¶¶ 92, 101.

Claim terms are given their ordinary and customary meaning, as would be understood by a person of ordinary skill in the art at the time of the invention, in light of the language of the claims, the specification, and the prosecution history. *See e.g., Hill-Rom Services, Inc. v. Stryker Corporation*, 755 F.3d 1367, 1371 (Fed. Cir. 2014). In applying this standard, there is “a ‘heavy presumption’ that a claim term carries its ordinary and customary meaning.” *Starhome GmbH v. AT&T Mobility LLC*, 743 F.3d 849, 857 (Fed. Cir. 2014).

Because of this presumption, a claim term may be construed contrary to its ordinary meaning only “under two circumstances: ‘(1) when a patentee sets out a definition and acts as [its] own lexicographer, or (2) when the patentee disavows the full scope of a claim term either in the specification or during prosecution.’” *Aventis Pharma S.A. v. Hospira, Inc.*, 675 F.3d 1324, 1330 (Fed. Cir. 2012); *Hill-Rom*, 755 F.3d at 1371. Because the intrinsic record includes no special definitions or disavowals for “block” or “subset” (and the Petition does not allege otherwise), “block” and “subset” should each be construed to have its ordinary meaning.

There is no dispute that in the field of the '628 Patent, a POSITA would understand that (1) a “texture block” is not an original image, it is a division of an image; and (2) a “subset” is not a block, it is a division of a block. Ex. 2001, ¶¶ 92, 101. Indeed, the evidence before the Board supports these—and only these—constructions.

For example, the only expert testimony before the Board on this issue is the declaration from Dr. Lastra, Patent Owner ATI’s proffered expert. Ex. 2001, ¶¶ 91-101. As Dr. Lastra explains and supports, a POSITA who had read the '628 Patent intrinsic evidence would have understood that a “texture block” is a division of an original texture image, not the image itself. Ex. 2001, ¶ 92. Dr. Lastra similarly explains and supports that a POSITA who had read the '628 Patent intrinsic evidence would have understood that a “subset” is a division of a texture block, not the block itself. Ex. 2001, ¶ 101. Petitioner argues that “block” is not limited to something equivalent to a 4x4 or 4x8 texel block of FXT1. Pet. at 55. Even so, Petitioner provided no expert opinion supporting the notion that the plain and ordinary meaning of “block” can comprise *an entire image* (as opposed to a block that is larger than 4x4 or 4x8 texels); indeed, the Petition and its supporting expert declaration simply ignore this issue. Even what Petitioner’s expert relies on for its so-called “broad” definition of block provides that: “[A] texture is *divided into a plurality of texture blocks.*” Ex. 1008 at [0075]; *id.* at [0049] (“Texture

Blocks: *Portions of a texture* that contain a plurality of texels.”); *see* Ex. 1003, ¶ 140 (citing Ex. 1008 at [0075]).

And contemporaneous documents support Dr. Lastra and Patent Owner ATI. Ex. 2001, ¶¶ 93-100. For example, FXTI itself calls the original image an “image” and confirms that texture blocks are a division of that original image: “FXT1™ texture compression works by *dividing an image into multiple 4x4 and/or 4x8 texel blocks.*” Ex. 1005 at 7. Similarly, Dr. Lastra explains that the seminal paper on GPU texture caching, by Hakura and Gupta, describes representing textures with “blocks” that are small enough size to fit in the texture caches of that time, for example, 2x2, 4x4, 8x8, and 16x16 texel blocks. Ex. 2001, ¶¶ 93-96. Meanwhile, other documents confirm these meaning, including the DXTC protocol, which likewise explains that “The *image* is broken up into 4x4 *blocks.*” Ex. 2001, ¶¶ 97-98. (quoting Ex. 2006 at 1 (emphasis added)).

And a POSITA who has reviewed the ’628 Patent’s intrinsic record could not possibly conclude that the ’628 Patent was using “texture block” to mean image and was using “subset” to mean texture block. Ex. 2001, ¶ 97. To the contrary, the ’628 Patent repeatedly confirmed that it was using block to refer to a subdivision of an original texture image, and subset to refer to a subdivision of a block. For example, the ’628 Patent describes that—just like in block-based prior

art such as DXTC and FXT1—in the present embodiments a block is a subdivision of an original texture image:

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 [texels], (e.g., 4x4 [texel] of 8x8 [texel] blocks) and compresses them.***

Initially, an original texture to be compressed is received in a texture compression engine. As with other block-based texture compression schemes, ***the original texture is then split or divided into original blocks of [texels]*** according to step 305.

Ex. 1001 at 6:1-4, 6:18-22, Fig. 3A.

The '628 Patent also explicitly confirmed and informed a POSITA that “subset” refers to the division of a block, not the division of an original texture image. For example:

Like DXTC, the basic compression algorithm implemented in a preferred embodiment divides or splits the image into discrete blocks of [texels], (e.g., 4x4 [texel] of 8x8 [texel] 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.

Thus, there can be no real dispute that a POSITA reading the intrinsic evidence would understand that the '628 Patent was using terms consistent with their ordinary meaning, such that (1) “texture” block is a division of an original

texture image, not the original texture image itself; and (2) “subset” is a division of a block and not the block itself. Thus, the Petition’s arguments that FXT1’s original “image” discloses and satisfies the claimed “texture image “ and that FXT1’s “texel blocks” discloses and satisfies the claimed “disjoint subsets” should be rejected as inconsistent with the proper claim construction and the “628 Patent’s use of those terms. Ex. 2001, ¶¶ 102-104.

b. *Treating FXT1’s “Image” as the Claimed “Texture Block” Would Render the Resulting System Inoperable*

As discussed, compression schemes divide an original image into smaller blocks because an original image is too large to fetch, receive, and decompress on demand as needed in graphics processing; instead, the original image is divided into smaller blocks which can be individually fetched, received and decompressed as needed. *See supra* Section II.C.2; Ex. 2001, ¶¶ 45-48, 92-96.

This is reflected in the challenged claims, which require “receiving a compressed texture **block**.” Ex. 1001 at Claim 1. But if FXT1’s original image were this compressed texture block as the Petition asserts, the resulting system would be inoperable. As Dr. Lastra explains, there are constraints in graphics processing that prevent operating on an entire image at once. Ex. 2001, ¶¶ 45-48, 92-96, 106-108. That is a fundamental reason for the existence of block-based

algorithms – if GPUs could make use of the entire image, there would be no need to divide images into blocks. *Id.* at ¶ 106.

GPUs are designed to load a portion of a texture image, for example, a 4x4 texel block, and not to load the entire image, for example, 2048x2048 texels. *Id.*, ¶ 107. This is consistent with FXT1’s disclosure, which is designed to handle 4x4 or 4x8 texel blocks. Ex. 1005 at 2, 5 (“FXT1™ texture compression works by dividing an image into multiple 4x4 and/or 4x8 texel blocks.”). 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. Ex. 2001, ¶ 108. 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.*; Ex. 1005 at 2. As noted previously, texture caches in 2006 would have less than 12 KB. Ex. 2005 (Doggett) at 139; Ex. 2001, ¶ 47.

Thus, the texture cache of the GPU would need to be greatly expanded in order to comply with Petitioner’s proposed modification to FXT1. Ex. 2001, ¶ 108. But modifying the GPU to make vast internal memory increases that would be necessary to handle the whole image would be impractical and costly. *Id.* Lastly, Petitioner fails to keep in mind that multiple textures are used at once, further taxing memory constraints. *Id.* Thus, Petitioner’s proposed modification is

simply impractical and would render a system using such a modification of FXT1 inoperable.

2. Petitioner's Obviousness Argument is Pure Hindsight Without Any Motivation

Recognizing the deficiencies in its main argument, Petitioner also argues 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, and then, the theory goes, one could call the newly theorized intermediate subdivision a “block” and call what FXT1 calls a block a “subset,” and therefore meet the requirements of the claims of the '628 Patent. Pet. at 55-57. Petitioner's theory is that one could pick some (apparently arbitrary) size that is a multiple of the 4x4 or 4x8 blocks actually disclosed in FXT1 so that this intermediate unit is then easily divisible back into those same 4x4 or 4x8 blocks. Pet. at 56-57 (“[A]n image may be partitioned into square blocks with each side having a length that is a multiple of 8 (8, 16, 24, etc.) . . . Each such square block may include two or more disjoint 4x4 and/or 4x8 texel blocks. Further, it would have been obvious to partition a 2048x2048 image into, for instance, 1,024 64x64 blocks, each of which may be further partitioned into a plurality of 4x4 and 4x8 disjoint subsets.”). This contorted and self-serving mapping to the claims has no basis in reality and is purely the use of hindsight to try to concoct something that maps to the claims. *See KSR International Co. v.*

Teleflex Inc., 550 U.S. 398, 422 (2007) (“A factfinder should be aware, of course, of the distortion caused by hindsight bias and must be cautious of arguments reliant upon *ex post* reasoning.”).

FXT1 provides no reason to create this additional theoretical level of subdivisions that conveniently divide back down into the blocks described in FXT1. Ex. 2001, ¶ 110. The Petition argues 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.” Pet. at 57. Petitioner makes no attempt to explain what this means or why it would “preserve the image fidelity of the texture” any more than what FXT1 already does, and simply cites to its expert witness declaration, which says the same thing verbatim, without any further technical explanation. Ex. 2001, ¶ 111. This is insufficient to prove obviousness. *See InTouch Techs., Inc. v. VGo Communs., Inc.*, 751 F.3d 1327, 1347, 1351 (Fed. Cir. 2014) (“Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning.”). Indeed, there is no need to “break up a large image into sets of data” — FXT1 already does that with the 4x4 or 4x8 blocks it describes. There is no motivation for adding this hypothetical intermediate unit. Furthermore, Petitioner fails to recognize that adding its additional level of

subdivisions would not solve any problems. Ex. 2001, ¶ 111. The utility of the '628 Patent is that subsets approximate contours of the texture. *Id.* Petitioner's proposed modification would still leave their "subsets" as squares of 4x4 and 4x8, providing no better quality. *Id.* Thus, a POSITA would not be motivated to modify FXT1 in this way.

VI. CONCLUSION

For any of the foregoing reasons, the Board should deny institution of the Petition.

Dated: July 27, 2023

Respectfully submitted,

/William A. Meunier/

William A. Meunier (Reg. No. 41,193)
Michael T. Renaud (Reg. No. 44,299)
Adam S. Rizk (Reg. No. 66,867)
MINTZ, LEVIN, COHN, FERRIS,
GLOVSKY AND POPEO, P.C.
One Financial Center
Boston, MA 02111
Telephone: 617-542-6000
Facsimile: 617-542-2241
E-mails: WAMeunier@mintz.com;
KCAmendt@mintz.com

Reza Dokhanchy (Reg. No. 62,795)
MINTZ, LEVIN, COHN, FERRIS,
GLOVSKY AND POPEO, P.C.
3580 Carmel Mountain Road, Suite 300
San Diego, CA 92130
Telephone: 858-314-1596
RDokhanchy@mintz.com

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. §42.24(d), Patent Owner hereby certifies, in reliance on the word count of the word-processing system (Microsoft Office Word 2010) used to prepare this preliminary response, that the number of words in this paper is 7,299. This word count excludes the tables of contents, table of authorities, certificate of word count, certificate of service, and table of exhibits.

Dated: July 27, 2023

/William A. Meunier/

William A. Meunier (Reg. No. 41,193)

CERTIFICATE OF SERVICE

I certify that a copy of Patent Owner's Preliminary Response and accompanying ATI Exhibits 2001 through 2008 is being served by electronic mail on the following counsel of record:

Lead Counsel

Jeffrey Johnson
Reg. No. 53,078
ORRICK, HERRINGTON &
SUTCLIFFE, LLP
609 Main Street, 40th Floor
Houston, TX 77002-3106
Telephone: (713) 658-6400
Facsimile: (713) 658-6401
Email: 3J6PTABDocket@orrick.com;
Realtek-AMD_OHS@orrick.com

Backup Counsel

Christopher J. Higgins
Reg. No. 66,422
ORRICK, HERRINGTON &
SUTCLIFFE, LLP
1152 15th Street, N.W.
Washington, DC 20005-1706
Telephone: (202) 339-8400
Facsimile: (202) 339-8500
Email: OCHPTABDocket@orrick.com;
Realtek-AMD_OHS@orrick.com

Steve Baik
Reg. No. 42,281
WHITE HAT LEGAL
P. O. Box 382
San Jose, CA 95002
Telephone: (650) 618-5282
Email: sbaik@whitehat.legal

Dated: July 27, 2023

/William A. Meunier/

William A. Meunier (Reg. No. 41,193)