

1
2
3
4
5
6
7
8
9
10
11

**UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA**

FUZZYSHARP TECHNOLOGIES INC.,

Plaintiff,

vs.

INTEL CORPORATION,

Defendant.

Case No.: 12-CV-04413 YGR

**AMENDED ORDER CONSTRUING CLAIM
TERMS IN DISPUTE AND GRANTING
SUMMARY JUDGMENT IN FAVOR OF
DEFENDANT INTEL CORPORATION**

12
13
14
15
16
17
18
19
20
21
22
23

Plaintiff Fuzzysharp Technologies, Inc. (“Fuzzysharp”) brings this patent infringement action against Defendant Intel Corporation (“Intel”) alleging that Intel HD Graphics Code “Sandy Bridge” infringes on U.S. Patent Nos. 6,172,679 (“the ‘679 Patent”) and 6,618,047 (“the ‘047 Patent”) owned by Fuzzysharp. Presently before the Court are two matters for decision: (1) the parties’ claim construction dispute; and (2) the motion of Intel (Dkt. No. 28), originally filed as a motion for judgment on the pleadings, on the grounds that the asserted claims are not patent-eligible under 35 U.S.C. § 101 because the asserted claims of U.S. Patent No. 6,618,047 (“the ‘047 Patent”) (claims 1, 8, 11, 12, 13, 46, 57, 64, 65, and 67) and of U.S. Patent No. 6,172,679 (“the ‘679 Patent”) (claims 1, 4, and 5) (the “Asserted Claims”)¹ are unpatentably abstract. The Court converted the motion to one for summary judgment, and permitted the parties to file supplemental briefing and evidence in support of and in opposition to the motion. (*See* Order Converting Motion for Judgment on the Pleadings Into Motion for Summary Judgment, Dkt. No. 45.)

24
25
26

Having carefully considered the papers submitted and the pleadings in this action, and for the reasons set forth below, the Court **ORDERS** as follows:

27
28

¹ At the time of the filing of Intel’s Motion, there were 40 asserted claims. Fuzzysharp subsequently withdrew all but these thirteen. (Dkt. No. 66-2, Exh. B [Declaration of Victoria Q. Smith in, 8/9/2013 email from Dave Fink to Victoria Smith].)

(1) The remaining terms² for construction have the following meaning:

<u>Term</u>	<u>Construction</u>
<p>Group 3('047 claim 1)</p> <p>“determining which of said at least one of 3-D surfaces or their sub-elements is always invisible or always visible”</p>	<p>“determining whether the 3-D surfaces are totally hidden or totally visible”</p>
<p>Group 4 ('679 claim 1)</p> <p>“determining for said viewpoint, whether each said selected surface is (a) an always unoccluded surface, an always hidden surface, or a remaining surface; or (b) an always unoccluded surface, or a remaining surface; or (c) an always hidden surface, or a remaining surface;”</p>	<p>“for each surface and from a predetermined viewpoint: (a) determining whether the surface is a totally visible surface, a totally hidden surface, or a surface that is neither totally visible or totally hidden; or (b) determining whether the surface is a totally visible surface or a surface that is neither totally visible or totally hidden; or (c) determining whether the surface is a totally hidden surface or a surface that is neither totally visible or totally hidden.”</p>

(2) The Asserted Claims are unpatentably abstract as a matter of law and Intel’s Motion for Summary Judgment is **GRANTED**.

BACKGROUND

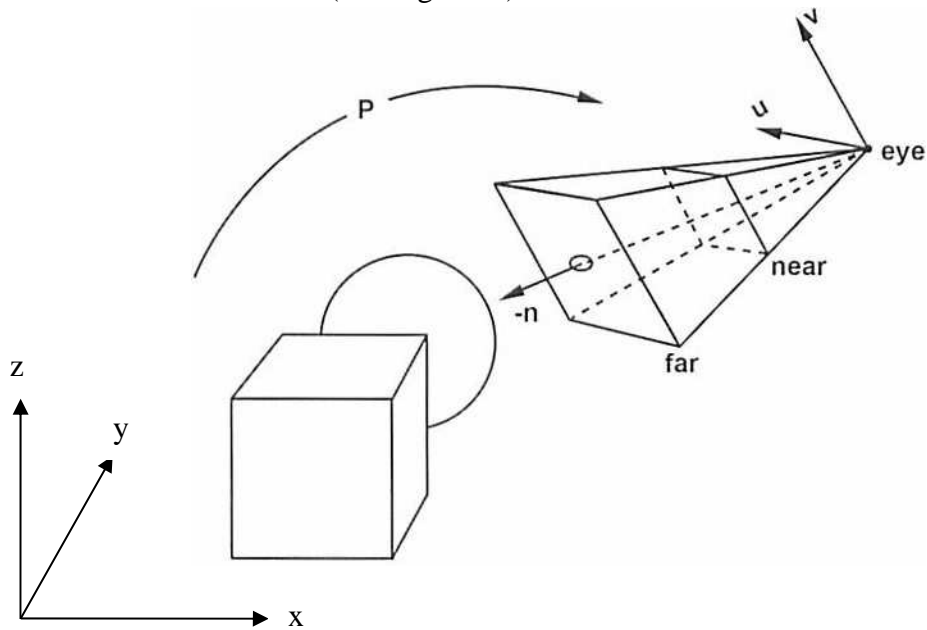
Fuzzysharp accuses Intel of infringing U.S. Patent Nos. 6,172,679 (“the ‘679 Patent”) and 6,618,047 (“the ‘047 Patent”). The patents claim methods to reduce the number of visibility computations required to render a three-dimensional scene as a two-dimensional digital computer graphic image. The patents describe methods of analyzing a scene before employing prior art

² Over the course of the claim construction briefing, the parties agreed on constructions for all terms at issue except Group 3 and Group 4.

1 rendering techniques. In essence, the patent claims describe a method of analyzing which parts of a
 2 scene will be visible and which will be hidden, and which may or may not be hidden/visible
 3 depending upon the vantage point of the viewer.

4 **I. THE TECHNOLOGY**

5 In 3D computer graphics, computers generate two-dimensional images to represent three-
 6 dimensional scenes. In general, a three-dimensional scene is first established by specifying the
 7 objects in the scene, how they are lit, and from which point(s) and which angle(s) they will be
 8 viewed (“viewpoints”). Producing two-dimensional images of that three-dimensional scene
 9 requires the specification of a viewpoint and viewing direction relative to the scene. For a given
 10 viewpoint and in scenes with opaque objects only, at most one opaque object or surface can be
 11 visible along any single line of sight or viewpoint. For example, in a scene where a cube sits
 12 behind a sphere from the perspective of the simulated viewer, the sphere would block the viewer
 13 from “seeing” some or all of the cube. (See Figure 1.)

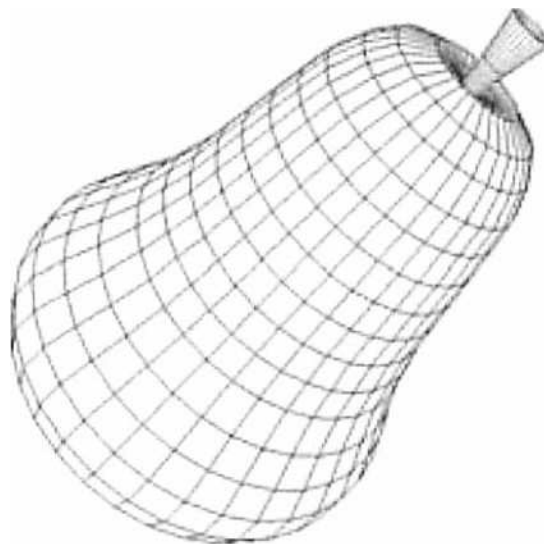


14
15
16
17
18
19 **Figure 1: A simulated viewpoint (upper right) viewing a geometric scene model (lower left)**

20
21
22
23
24
25 Generally the next steps in rendering a three-dimensional scene as a two-dimensional image
 26 are that the properties of the display device are specified, and the coordinates for display of the
 27 image on the screen of the device, *i.e.* the “pixels” (points of light with properties that may include
 28 color and intensity). The display screen contains a two dimensional array of pixels, each of which

1 displays a color computed for some point in the scene. The pixels collectively form the images
2 displayed on the screen. In order to determine which surface should appear in an image from a
3 given viewpoint, the relative “depth” in the scene (*i.e.*, location on the z-axis) for the various object
4 surfaces must be determined and compared. If objects are opaque and aligned such that one object
5 “blocks” the view of another, then the object that is “nearer” to the viewpoint (*i.e.*, less “depth” in
6 the scene) will have visible surfaces, while the object that is behind the first and “farther” from the
7 viewpoint (*i.e.*, greater “depth” in the scene) will have surfaces hidden by the first.

8 3D graphics are typically rendered using GPUs (graphics processing units). The surfaces
9 that will appear in the image from a particular viewpoint are represented as a collection of basic
10 geometric shapes, or “primitives.” (*See* Figure 2.) A GPU is specialized hardware that receives the



11
12
13
14
15
16
17
18
19
20
21 **Figure 2: Collection of Primitives For Displaying Visible Surfaces of an Object From a**
22 **Particular Viewpoint**

23 primitives or surfaces from an application program and then displays the primitives or surfaces as
24 an image on the computer screen. Upon receiving a primitive, the GPU converts it into a group of
25 pixels that represents the points on a display device.

26 GPUs typically use frame buffers, which are a portion of computer memory (buffer)
27 organized into a discrete grid (frame) of pixels that store information about each pixel. In the frame
28 buffer, each pixel has memory bits dedicated to representing the color and light intensity to be

1 displayed at the location corresponding to that pixel. The frame buffer is connected to specialized
2 computer hardware that can rapidly scan out the contents of memory to determine what should be
3 displayed.

4 Many computer graphics rendering systems also include a specialized portion of computer
5 memory designated as a “z-buffer” or “depth buffer.” The GPUs can use the z-buffer to store data
6 about the “depth” for each pixel for the purpose of eliminating hidden surfaces at the level of
7 individual pixels or for groups of pixels. The process of using the z-buffer to determine which
8 scene points are visible is known as “z-buffering” or “z-testing.” For scene points that are
9 determined to be visible, the values for color and other properties will be written into the frame
10 buffer and become the values for the corresponding pixel. For a scene point that is determined to
11 be hidden, the values for the color and other properties of the point will not be written into the
12 frame buffer, and the data for the pixel that corresponds to that point will not be overwritten. Thus,
13 the frame buffer and z-buffer are used together to determine what should be displayed on the
14 screen.

15 **II. THE PATENTS**

16 The patents-in-suit involve 3D computer graphics and purport to disclose “an improved
17 method for performing visibility calculations.” (‘679 Patent Title, Field of Invention, col. 2:17- 20.)
18 The patents describe the background art, and the problem to be solved by the patents, in part as
19 follows:

20 Visible surface detection is one of the most basic operations in 3D graphics. It
21 is applied to generate images of surfaces directly visible to a viewer. . . . The
22 standard strategy of visible surface detection is to divide surfaces into patch
23 elements and compare the spatial relationship between these elements. Using
24 this strategy, the visibility of surfaces cannot be determined until they have been
25 analysed in detail. Although many techniques have been developed to address
26 this issue, none are ideal as they either still require elaborate analysis of
27 surfaces, or they impose various restrictions to the scene. The limitations of the
28 current techniques can seriously affect the speed of visible surface computation.
If the scene is complicated, many surfaces may be invisible. However, the
image generation is often slowed down by the need to analyse each surface
element in detail.

(‘047 Patent, col. 1:18-36.)

1 The '047 Patent is a continuation of the '679 Patent and shares a nearly identical
2 specification. The only difference between the patents lies in the Summary of the Invention, which
3 was modified in the '047 Patent to reflect differences in the claims in the patents. The Asserted
4 Claims (claims 1, 4, and 5 of the '679 Patent, and claims 1, 8, 11, 12, 13, 46, 57, 64, 65, and 67 of
5 the '047 Patent) describe a method (and variations) for reducing the number of visibility
6 computations a GPU is required to perform in order to render a three-dimensional computer graphic
7 image. The method reduces computations by pre-processing the surfaces from a group of
8 viewpoints by: predetermining the visibility of surfaces from a group of viewpoints; creating data
9 structures storing the resulting visibility computations; and then ignoring the pre-identified surfaces
10 during later visibility computations, such as the visibility computations performed during rendering.

11 Claim 1 of the '047 Patent is illustrative:

- 12 1. A method of reducing the visibility related computations in 3-D computer
13 graphics, the visibility related computations being performed on 3-D surfaces
14 or their sub-elements, or a selected set of both, the method comprising:
15 identifying grid cells which are under or related to the projections or extents of
16 projections associated with at least one of said 3-D surfaces or their sub-
17 elements;
18 comparing data associated with said at least one of 3-D surfaces or their sub-
19 elements with stored data associated with the grid cells;
20 determining which of said at least one of 3-D surfaces or their sub-elements is
21 always invisible or always visible to a viewpoint or a group of viewpoints
22 by projection based computations prior to a visibility computation; and
23 ignoring said determined at least one of the 3-D surfaces or their sub-elements
24 during said visibility computation.

25 ('047 Patent, col. 27:66-28:17.) The claim language refers to pre-processing the visibility of
26 surfaces (or parts of surfaces) before traditional rendering ("prior to a visibility computation"). The
27 claimed method creates a set of fixed, imaginary planes onto which the surfaces or parts of surfaces
28 of the objects in the scene are projected ("projections"), imposes on each plane an imaginary grid
consisting of abstract individual cells ("grid cells"), creates a data structure that stores depth data
for each abstraction ("stored data associated with the grid cells"), and then compares the data for a
given surface with the data for the corresponding grid cell to determine whether that surface is
totally invisible or totally visible. After such pre-processing, an application program can identify

1 which surfaces are totally invisible or totally visible from a given viewpoint group by consulting
2 the data structures associated with the viewpoint group. The application program can then “ignore”
3 (‘047 Patent claim 1), “skip” (‘047 Patent claims 11 and 12), or “exempt” (‘679 Patent claim 1)
4 totally invisible surfaces when sending data to the GPU for rendering an image. Alternatively, the
5 program could send data for totally visible (“always unoccluded”) surfaces to the GPU for
6 rendering but could “ignore,” “exempt,” “skip,” or otherwise direct the GPU not to perform the z-
7 buffering tests for such surfaces. (‘679 Patent claim 1 and ‘047 Patent claims 1, 46.)

8 The written description likewise describes pre-processing surfaces and for each viewpoint
9 group: classifying the surfaces into categories, creating data structures storing that classification,
10 and then using the stored data to exempt totally visible and totally invisible surfaces from further
11 visibility computations. (*See* ‘679 Patent col. 14:13-25 [“In subsequent rounds of computations,
12 where the form factors between patches have to be re-evaluated, the fuzzy projection computations
13 are not repeated. The totally visible/non-hiding and the totally invisible patches of each-viewpoint
14 can be simply retrieved from the stored information”].)

15 The overarching goal of the patents is to “reduc[e] the complexity of visibility calculations
16 required for the production of multi-dimensional computer generated images.” (‘679 Patent col.
17 28:25-27.) The parties have agreed that this means “decreasing the number of computational
18 operations to perform visibility computations.” (Dkt. No. 64 at 5.) By analyzing a scene from a
19 viewpoint or group of viewpoints at the outset and creating data structures reflecting the visibility
20 of surfaces or portions of surfaces rather than individual pixels or groups of pixels, the claimed
21 method was designed to reduce the portions of a scene that had to be subjected to the full-blown
22 visibility and rendering computations that were known and used in the prior art.

23 CLAIM CONSTRUCTION

24 For purposes of claim construction, the sole dispute left between the parties is whether the
25 claim terms require a 3-part categorization (*i.e.*, totally hidden, totally visible, and remaining) or
26 whether they are satisfied by a 2-part determination only (*i.e.*, totally hidden vs. remaining, totally
27 visible vs. remaining, or totally visible vs. totally hidden).

28

1 Both Group 3 and Group 4 claims concern categorization of surfaces for pre-processing.
2 The parties dispute whether the claims require a three-part determination or a two-part
3 determination. The parties competing constructions follow, with the differences in bold and italics:

<u>Term</u>	<u>Fuzzysharp’s Proposed Construction</u>	<u>Intel’s Proposed Construction</u>
<p>4 Group 3 5 “determining which of said 6 at least one of 3-D surfaces 7 or their sub-elements is 8 always invisible or always 9 visible” 10 11 (‘047 claim 1)</p>	<p>“determining which of the 3-D surfaces are <i>either</i> hidden or visible <i>from a single viewpoint, or a group of viewpoints.</i>”</p>	<p>“determining whether the 3-D surfaces are totally³ hidden or totally visible”</p>
<p>12 Group 4 13 “determining for said 14 viewpoint, whether each 15 said selected surface is 16 (a) an always unoccluded 17 surface, an always hidden 18 surface, or a remaining 19 surface; or 20 (b) an always unoccluded 21 surface, or a remaining 22 surface; or 23 (c) an always hidden 24 surface, or a remaining 25 surface;” 26 27 (‘679 claim 1)</p>	<p>“for each surface and from a predetermined viewpoint, <i>carrying out one of the following groups of tests in (a), or (b), or (c) to determine, respectively:</i></p> <p>(a) <i>which single one of the following categories the surface falls:</i> a totally visible surface, a totally hidden surface, or a surface that is neither totally visible or totally hidden; (b) <i>which single one of the following categories the surface falls:</i> a totally visible surface or a surface that is neither totally visible or totally hidden; or (c) <i>which single one of the following categories the surface falls:</i> a totally hidden surface or a surface that is neither totally visible or totally hidden.”</p>	<p>“for each surface and from a predetermined viewpoint: (a) <i>determining whether the surface is</i> a totally visible surface, a totally hidden surface, or a surface that is neither totally visible or totally hidden; <i>or</i> (b) <i>determining whether the surface is</i> a totally visible surface or a surface that is neither totally visible or totally hidden; <i>or</i> (c) <i>determining whether the surface is</i> a totally hidden surface or a surface that is neither totally visible or totally hidden.”</p>

28 ³ Although Fuzzysharp’s proposed construction for claim 1 of the ‘047 patent does not indicate that the terms “hidden” and “visible” mean “totally hidden” and “totally visible,” the parties have agreed that they do. (Dkt. No. 64 at 1-2.)

1 Thus, when Claim 1 of the '679 Patent recites:

2 ...for selected ones of said surfaces, determining for said viewpoint whether
3 each said selected surface is

4 (a) an always unoccluded surface, an always hidden surface, or a remaining
5 surface; *or*

6 (b) an always unoccluded surface, or a remaining surface; *or*

7 (c) an always hidden surface, or a remaining surface; ...

8 ('679 Patent, col. 28:33-39), the parties agree that the italicized "or's" are disjunctive, such that
9 only one of the three sets of tests ((a), (b), and (c)) need be performed. Their dispute lies within
10 each of (a), (b), and (c).

11 Under Intel's construction, the method of the patents pre-processes the scene to determine
12 from a viewpoint or group of viewpoints which surfaces of the scene are totally hidden, which are
13 totally visible, and which are "remaining." Intel thus proposes that "remaining" means "neither
14 totally hidden nor totally visible" wherever it is used in the patents. Using those data structures, a
15 program may then avoid sending the totally hidden surfaces to the graphics processing unit
16 ("GPU") for rendering, since a totally hidden surface need not be rendered at all. Similarly, the
17 program can send the totally visible surfaces to the GPU, but instruct that no further visibility
18 computations need to be performed to render the surface. Thus, the method of the patents tests for
19 two categories of surfaces, totally hidden or totally visible, and any surface not meeting either of
20 those categories falls into a third, "remaining surfaces." The remaining surfaces are those surfaces
21 that will need additional visibility computations performed by the GPU prior to or during rendering.
22 According to the patents, this categorization of surfaces reduces the number of visibility
23 computations that the GPU must perform, leading to an overall gain in system efficiency.

24 According to Fuzzysharp, Intel's interpretation of the term "remaining" is inaccurate and
25 leads to confusion. The patents describe different possible operations that test whether surfaces are
26 totally visible or totally invisible. The simpler operations described test only for one type of
27 surface. For example, '679 Patent, Claim 1(c) tests only for hidden surfaces, and under that test a
28 surface is either totally hidden or not. Under this test, Fuzzysharp contends, the "not" is the
"remaining surface." Likewise, the operation described in the '679 Patent, Claim 1(b) is to test for
totally visible (or unoccluded) surfaces. The nature of this test is that, if the surface fails the test for

1 being visible, then it is classified as being a “remaining” surface, in this case one that can be
2 described as either: (1) totally hidden; or (2) neither totally hidden nor totally visible. Under ‘679
3 Patent Claim 1(a) tests are carried out for both totally visible *and* totally hidden surfaces, and
4 failing both tests also means that the surface is “remaining.” However, in this scenario “remaining
5 surfaces” would *only* be neither totally hidden nor totally visible. Fuzzysharp further contends that
6 the ‘047 Patent, Claim 1 describes only two simple operations: a test for always hidden (invisible)
7 surfaces *or* always visible surfaces; those surfaces failing the test used are classified as remaining
8 surfaces. In Fuzzysharp’s reading of the claim language, a “remaining” surface is a simply a by-
9 product of steps for determining the information necessary to simplify computations, and therefore
10 means different things depending upon the test used.

11 **I. CLAIM LANGUAGE**

12 Unlike Fuzzysharp’s proposed construction, Intel’s construction gives meaning to the
13 limitations in the claims themselves. First, looking at the disputed portion of the claim in context
14 supports giving a single, consistent meaning to the term “remaining surface” as a category distinct
15 from both “hidden” and “visible/unoccluded.” The language of Claim 1 recites, in part:

16 ...determining for said viewpoint whether each said selected surface is
17 (a) an always unoccluded surface, an always hidden surface, or a remaining
18 surface; or
19 (b) an always unoccluded surface, or a remaining surface; or
20 (c) an always hidden surface, or a remaining surface;
21 *wherein said remaining surface is a surface which is unable to be determined*
22 *with certainty as to whether it is either unoccluded or hidden*

23 (‘679 Patent, col. 28:33-42, emphasis added.) Thus, the “wherein” clause is written to apply to
24 *each* of (a), (b), and (c), and to provide a definition of a “remaining surface” for all, not just a
25 definition that applies only to (a). When a “wherein” clause “expresses the inventive discovery,”
26 rather than just stating a result of certain limitations, it is definitional. *See Griffin v. Bertina*, 285
27 F.3d 1029, 1034 (Fed.Cir. 2002); *Intergraph Hardware Technologies Co. v. Toshiba Corp.*, 508 F.
28 Supp. 2d 752, 769 (N.D. Cal. 2007) (“wherein” clause stated limitations that were part of
construction rather than expressing result of limitation).

1 The other limitations of Claim 1 are consistent with Intel’s reading. First, Claim 1 requires
2 “exempting from said occlusion or invisibility relationship computation those surfaces which are
3 *either* always unoccluded *or* always hidden.” (‘679 Patent col. 28:43-45, emphasis added). The
4 “exempting” limitation can be met only if a surface has been tested sufficiently to determine
5 whether it is always visible or always hidden, rather than testing for only one of these categories, as
6 Fuzzysharp suggests. Second, Claim 1 defines “*said remaining surfaces*” in the “wherein” clause
7 and then goes on to require “maintaining a record of *said remaining surfaces*.” (‘679 Patent col.
8 28:40-42, 46.) The “maintaining” limitation cannot be met unless surfaces have been tested
9 sufficiently to determine those that are *neither* always visible nor always hidden and thus fall into
10 the third category, “remaining,” as to which a record is to be maintained.

11 Fuzzysharp’s proposed construction makes ‘679 Patent Claim 1 clauses (b) and (c)
12 redundant of clause (a). Fuzzysharp proposes that ‘679 Patent Claim 1(a) be construed to mean
13 carrying out a test to determine into which *single* one of the following categories the surface falls: a
14 totally visible surface, *or* a totally hidden surface, *or* a surface that is neither totally visible or
15 totally hidden. In other words, a test under Claim 1(a) tests for only one condition, with a binary
16 result (*i.e.*, meets category or does not meet category). Under that construction, however, clauses
17 (b) and (c) would have no meaning, because they would not cover a case not already expressed in
18 clause (a).

19 Intel’s construction gives effect to all three clauses since clause (a) tests for all three
20 categories (totally visible, totally hidden, and “remaining”), clause (b) tests for two of the three
21 categories (totally visible and “remaining,” with all others being totally hidden), and clause (c) tests
22 for a different set of two of the three (totally invisible and “remaining,” with all others being wholly
23 visible). Thus, Intel’s construction fits with the language of ‘679 Patent Claim 1 itself while
24 Fuzzysharp’s does not.

25 Though the ‘047 Patent claims do not use the term “remaining surfaces,” the same
26 categorization is implicit because those claims simply require determining the totally hidden and
27 totally visible surfaces and then exempting *both* those categories of surfaces from future visibility
28 computations by the GPU. In both cases, the GPU must then perform visibility computations only

1 on the “remaining surfaces.” The clause in dispute requires “determining which of said at least one
2 of 3-D surfaces or their sub elements is always invisible or always visible.” But the final step
3 requires “ignoring said determined at least one of the 3-D surfaces or their sub-elements during said
4 visibility computations.” (‘047 Patent, col. 28:9-15.) The claim thus requires testing to determine
5 which surfaces are “always invisible or always visible” so that both wholly hidden and wholly
6 visible surfaces are excluded. Read in context, the “or” does not suggest that only one test need be
7 performed; it instead signifies that a surface cannot be both “always invisible” and “always
8 visible.”

9 In sum, in order for the claim language itself to be internally consistent, the language in
10 both the ‘679 Patent, Claim 1, and the ‘047 Patent, Claim 1 require a construction that means the
11 method can test in such a way as to result in three categories of surfaces, not just two.

12 **II. PATENT SPECIFICATION**

13 Fuzzysharp’s position is, essentially, that reading the patent to require determining both
14 “always visible” and “always invisible” for all surfaces would not result in the reduction of
15 visibility-related computations that is part of the goal of the patent, as expressed in the Summary of
16 Invention for both patents. (‘679 Patent, col. 2: 22-65, ‘047 Patent, col. 2:22-46.) However, Intel
17 correctly notes that the patent specification in the ‘679 Patent itself states:

18 In the first round of computations, all the fuzzy projection and normal hemicube
19 computations for every receiving patch are carried out as required. ***During***
20 ***these computations, the emission patches at each group/viewpoint are***
21 ***classified into three groups: totally visible/non-hiding, totally invisible, and***
the remainder. This classification information is stored using data structures
such as arrays or linked lists.

22 (‘679 Patent, col. 14:13-19, emphasis added; *see also* ‘047 Patent ,col. 13:46-50 [same]).

23 Elsewhere, the specification similarly describes testing for both total invisibility and total visibility,
24 and all others into a “remaining” category:

25 “[a]fter the filtering off of totally invisible patches by the invisibility technique,
26 the visibility technique detects totally visible and non-hiding patches. ... The
27 remaining patches are not totally invisible. They are also not totally
28 visible/non-hiding.”

1 ('679 Patent, col. 14:59-65; *see also* '047 Patent, col. 14:28-29 [same].) Thus the specification
2 supports a construction requiring that the claims will, at least in some cases, put surfaces into three
3 categories. Contrary to Fuzzysharp's argument, such a construction is also consistent with the goal
4 of eliminating a full visibility computation for surfaces already known to be wholly visible or
5 wholly hidden. Thus the specification supports Intel's construction reading the claim language in
6 both the '679 Patent, Claim 1, and the '047 Patent, Claim 1 to result in a determination of three
7 categories of surfaces.

8 **III. PROSECUTION HISTORY**

9 While not necessary to the Court's construction determination, Intel's proposed construction
10 is consistent with the prosecution history stated in the patents. The patent examiner initially
11 rejected the claim language in the '047 Patent which stated "determining which of the 3-D surfaces
12 or their sub-elements are always invisible or always visible." (Dkt. No. 66-2 to 66-6 [Declaration
13 of Victoria Q. Smith], Exh. C ['047 Patent File History, Application] at 48.) The patent examiner
14 found that the claim was obvious in light of the prior art in *Kadota*, which disclosed a
15 determination whether 3-D surfaces were always visible or invisible from a viewpoint. The
16 rejection stated:

17 *As per independent claim 1, Kadota discloses determining 3D surfaces that*
18 *are always visible or invisible* to a viewpoint by projection (abstract) and
19 treating those surfaces differently from remaining surfaces ..., it would have
20 been obvious to one of ordinary skill in the art at the time of invention to use the
21 disclosure of Kadota because he teaches transmitting and processing
22 predetermined projection coordinates to a hidden surface processor *which*
evaluates both visible and invisible surfaces, transmits the visible surfaces and
ignores the invisible surfaces.

(Smith Dec., Exh. E ['047 Patent File History, September 22, 2000 Rejection] at 2, emphasis
23 added.)

24 Previously, the Patent and Trademark Office had allowed the '679 Patent, the Reasons for
25 Allowance having stated:

26 the applicant claims sorting surfaces as visible, hidden or remaining (either
27 visible or nonvisible) and processing the remaining surfaces to perform and
28

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

reduce visible/hidden surface computations, which is not disclosed in the prior art of record.

(Smith Dec., Exh. D [‘679 Patent File History, November 15, 1999 Notice of Allowance] at 2.) Fuzzysharp does not address this prosecution history or offer any contradictory arguments.

Thus, based on the prosecution history offered by Intel, the Patent and Trademark Office appears to have considered the three categories of surfaces to be critical in distinguishing the patents over prior art.

THE COURT’S CONSTRUCTION:

Based upon the foregoing, the Court interprets the claims at issue as follows:

Term	Construction
Group 3(‘047 claim 1) “determining which of said at least one of 3-D surfaces or their sub-elements is always invisible or always visible”	“determining whether the 3-D surfaces are totally hidden or totally visible”
Group 4 (‘679 claim 1) “determining for said viewpoint, whether each said selected surface is (a) an always unoccluded surface, an always hidden surface, or a remaining surface; or (b) an always unoccluded surface, or a remaining surface; or (c) an always hidden surface, or a remaining surface;”	“for each surface and from a predetermined viewpoint: (a) determining whether the surface is a totally visible surface, a totally hidden surface, or a surface that is neither totally visible or totally hidden; or (b) determining whether the surface is a totally visible surface or a surface that is neither totally visible or totally hidden; or (c) determining whether the surface is a totally hidden surface or a surface that is neither totally visible or totally hidden.”

//
//

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

SUMMARY JUDGMENT

Intel brings its motion for summary judgment on the grounds that, pursuant to 35 U.S.C. § 101, the Asserted Claims are not patent-eligible. Intel argues that the Asserted Claims set forth an abstract idea without adding more than an instruction to “apply it,” or recite an abstract idea without requiring any application containing an “inventive concept,” or both. Whether the claims satisfy 35 U.S.C. § 101’s criteria for patent-eligible subject matter is a question of law for the Court. *Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1333 (Fed. Cir. 2012). The Federal Circuit has held that “any attack on an issued patent based on a challenge to the eligibility of the subject matter must be proven by clear and convincing evidence.” *Ultramercial, Inc. v. Hulu, LLC*, 722 F.3d 1335, 1342 (Fed. Cir. 2013).

Here, there are no disputed issues of fact raised by the parties.⁴ The parties agreed on constructions for all of the claim terms in dispute but two. Those two terms were the subject of claim construction briefing and argument, and their constructions are set forth above, in the first part of this Order.

I. LEGAL FRAMEWORK

In analyzing patentability under Section 101, a court must first identify whether the claimed invention fits within one of the four statutory classes set forth in the statute. Section 101 specifies four broad, independent categories of inventions or discoveries that are patent eligible: “process, machine, manufacture or composition of matter.” 35 U.S.C. § 101. Second, a court must assess whether the claimed invention falls into any of the three specific exceptions established by Supreme Court precedent: “laws of nature, physical phenomena, and abstract ideas.” *Diamond v Chakrabarty*, 447 U.S. 303, 309 (1980). These exceptions have grown out of the Supreme Court’s

⁴ Intel initially brought this motion as one for judgment on the pleadings. (Dkt. No. 28.) Based upon the arguments made in opposition by Fuzzysharp, the Court converted the motion to one for summary judgment and permitted the parties to file supplemental briefing and evidence, and specifically directed that the parties identify any terms or phrases that required a claim construction prior to a decision on summary judgment. (Dkt. No. 45.)

1 “concern that patent law not inhibit further discovery by improperly tying up the future use of laws
2 of nature,” natural phenomena, and abstract ideas. *Mayo Collaborative Servs. v. Prometheus Labs.,*
3 *Inc.*, 132 S. Ct. 1289, 1301 (2012). “A principle, in the abstract, is a fundamental truth...[which]
4 cannot be patented.” *Gottschalk v. Benson*, 409 U.S. 63, 67 (1972) (internal citations and
5 quotations omitted) (“*Benson*”). “Phenomena of nature, though just discovered, mental processes,
6 and abstract intellectual concepts are not patentable, as they are the basic tools of scientific and
7 technological work.” *Id.* Included within the exception for abstract ideas are such concepts as
8 mathematical formulae. *Id.* at 68.

9 A court must then determine whether the patent claims “pose any risk of preempting an
10 abstract idea.” *CLS Bank Int’l v. Alice Corp. Pty. Ltd.*, 717 F.3d 1269, 1282 (Fed. Cir. 2013). Even
11 if the patent claims rely on an abstract concept such as a mathematical formula, the “balance of the
12 claim can be evaluated to determine whether it contains additional substantive limitations that
13 narrow, confine, or otherwise tie down the claim so that, in practical terms, it does not cover the
14 full abstract idea itself.” *Id.* at 1282 (citing *Mayo*, 132 S.Ct. at 1300; *Bilski v. Kappos*, 130 S. Ct.
15 3218, 3231 (2010); and *Diehr*, 450 U.S. at 187).

16 **A. Supreme Court Authorities**

17 Over time, the Supreme Court has set forth a number of different factors to consider in
18 distinguishing a patentable invention from an unpatentable abstract concept. Five Supreme Court
19 cases are particularly instructive here. First, in *Benson*, the Supreme Court held that a
20 mathematical formula for converting binary-coded decimal numerals into pure binary code *was not*
21 patentable because the process claimed was “so abstract and sweeping” that it could cover usage in
22 a variety of operational settings and could be performed through any existing or future machinery,
23 or without machinery at all. *Benson*, 409 U.S. at 67. The fact that the process or method is carried
24 out with the use of common computer components, such as computer memory, does not render an
25 otherwise abstract concept patentable and processes that “can be carried out in existing computers
26 long in use, no new machinery being necessary” are ineligible for patent protection. *Id.* at 67.

27 Following thereafter, in *Parker v. Flook*, 437 U.S. 584 (1978) (“*Flook*”), the Supreme Court
28 held that a patent which described a mathematical formula for calculating alarm limit values *was*

1 *not* patentable due to the abstract nature of the subject matter. There, the patentee sought to
2 distinguish *Benson* by arguing that, although there were a number of “post-solution” uses for the
3 formula, the patent included specific “post-solution” activity to make the formula useful in the
4 petrochemical and oil-refining industries. The Supreme Court rejected this distinction, stating that

5 [t]he notion that post-solution activity, no matter how conventional or obvious
6 in itself, can transform an unpatentable principle into a patentable process exalts
7 form over substance. A competent draftsman could attach some form of post-
8 solution activity to almost any mathematical formula; the Pythagorean theorem
9 would not have been patentable, or partially patentable, because a patent
application contained a final step indicating that the formula, when solved,
could be usefully applied to existing surveying techniques.

10 *Flook*, 437 U.S. at 590 (1978). Thus, while a claimed invention does not lose patent eligibility
11 simply because it relies on a phenomenon of nature or mathematical formula, such a phenomenon
12 or formula cannot, on its own, be patentable without “some other inventive concept in its
13 application.” *Id.* at 594.

14 Subsequently, by contrast, the Supreme Court found patentable under Section 101 a process
15 combining an abstract formula with practical applications and limitations in *Diamond v. Diehr*, 450
16 U.S. 175 (1981) (“*Diehr*”). There, the patent claimed a process for curing synthetic rubber that,
17 while involving an abstract mathematical formula, used that formula in combination with a novel
18 set of steps to be followed in a particular application. The overall process was patentable because it
19 integrated the equation into the rubber curing process as a whole, using constant temperature
20 readings inside the mold, and making repeated recalculations to determine the optimum time to
21 open the rubber press. *Id.* The Court held that patent avoided the abstractness exception to patent
22 eligibility since the patentees did not “seek to pre-empt the use of that equation. Rather, they seek
23 only to foreclose from others the use of that equation *in conjunction with all of the other steps* in
24 their claimed process.” *Id.* at 187 (emphasis added). Thus an abstract idea, law of nature, or
25 mathematical formula could not be patented but “an *application* of a law of nature or mathematical
26 formula to a known structure or process may well be deserving of patent protection.” *Id.*

27 In its 2010 *Bilski* decision, the Supreme Court reiterated these principles. *Bilski*, 130 S. Ct.
28 at 3231. The Court held that the Federal Circuit’s “machine-or-transformation test,” while not

1 dispositive of patent eligibility, provides a “useful tool” for determining whether a claimed method
2 or process is patent eligible. *Bilski*, 130 S. Ct. at 3226-27. Under that formulation, a method or
3 process is patent eligible when: “(1) it is tied to a particular machine or apparatus, or (2) it
4 transforms a particular article into a different state or thing.” *Id.* at 3225 (quoting *In re Bilski*, 545
5 F.3d 943, 954 (Fed. Cir. 2008) (*en banc*)). In *Bilski*, the Court found that claims describing the
6 concept of hedging and a mathematical formula for hedging against risk in price fluctuations for
7 buyers and sellers of commodities were not patentable. The Court held that “hedging” was a basic
8 economic concept, and reducing the concept to a mathematical formula did not take it out of the
9 realm of abstract ideas. *Id.* Likewise, adding claims that simply applied the formula to particular
10 markets was not enough to convert the abstract concept into a patentable one, just as adding a field
11 of use or common post-solution components was not sufficient in *Flook*. *Id.*

12 Most recently, the Supreme Court held that claims were too abstract, and not patentable
13 under section 101, where “the claimed processes (apart from the natural laws themselves) involve
14 well-understood, routine, conventional activity previously engaged in by researchers in the
15 field...” *Mayo*, 132 S. Ct. at 1294. The Court held that a claim must apply an abstract idea in a
16 way that reflects an “inventive concept,” which it illustrated by distinguishing its decisions in *Flook*
17 and *Diehr*. In *Flook*, the claimed method used a computer to calculate alarm-limit values to signal
18 dangers in operating a catalytic converter, but the process did “nothing other than ‘provid[e] an
19 unpatentable formula for computing [the] updated alarm limit.” *Mayo*, 132 S. Ct. at 1299 (quoting
20 *Flook*, 437 U.S. at 586) (internal quotations and brackets omitted). All the other steps—the
21 chemical processes involved in catalytic conversion, the monitoring of chemical process variables,
22 the use of alarm limits to signal danger, and the adjusting of those limits through the use of
23 computers for “automatic monitoring-alarming”—were “‘well known.’” *Id.* at 1299 (quoting
24 *Flook*, 437 U.S. at 594). Thus, “putting the [abstract] formula to the side, there was no ‘inventive
25 concept’ in the claimed application of the formula.” *Id.* In *Diehr*, by contrast, the “other steps”
26 added something “that transformed the process into an inventive application of the formula.”
27 *Mayo*, 132 S Ct. at 1299 (process was patentable because the other steps, and the combination of
28 them, were not “obvious, already in use, or purely conventional”).

1 **B. Recent Federal Circuit Decisions**

2 The Federal Circuit has applied these principles in a recent cases involving computing
3 technologies. In one, the Federal Circuit found unpatentable a claimed method which disclosed a
4 number of computer components needed to carry it out. *Accenture Global Servs., GmbH v.*
5 *Guidewire Software, Inc.*, 728 F.3d 1336, 1343 (Fed. Cir. 2013) (components required included
6 “CPU, ROM, RAM, I/O Adapter, Communication Adapter, Display Adapter, and a User Interface
7 Adapter”). The listed computer components constituted what was described in the patent as a
8 “representative hardware environment.” *Id.* The Federal Circuit found such components
9 insufficient to provide meaningful limitations on the abstract concept of the patent itself. *Id.*
10 “[S]imply implementing an abstract concept on a computer, without meaningful limitations to that
11 concept, does not transform a patent-ineligible claim into a patent-eligible one.” *Id.* at 1345 (citing
12 *Bancorp*, 687 F.3d at 1280). “[T]he complexity of the implementing software or the level of detail
13 in the specification does not transform a claim reciting only an abstract concept into a patent-
14 eligible system or method.” *Id.*

15 *Accenture* echoed other recent Federal Circuit decisions finding that simply adding a
16 storage or calculation step performed by a digital computer component did not limit an otherwise
17 abstract concept sufficiently to make it patentable. *See Bancorp Servs., L.L.C. v. Sun Life Assur.*
18 *Co. of Canada (U.S.)*, 687 F.3d 1266, 1278 (Fed. Cir. 2012) (computer and “high density
19 removable storage means ... such as a compact disc” required by patent claims used only for basic
20 computing functions and repetitive calculations did not impose meaningful limits on the scope of
21 those claims to make them patentable); *Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1333 (Fed. Cir.
22 2012) (“simply adding a ‘computer aided’ limitation to a claim covering an abstract concept,
23 without more, is insufficient to render the claim patent eligible”); *Cybersource Corp. v. Retail*
24 *Decisions, Inc.*, 654 F.3d 1366, 1376 (Fed. Cir. 2011) (added computer storage step naming
25 particular type of storage component does not render otherwise unpatentable process patent-
26 eligible) (citing *Benson*, 409 U.S. at 73-74).

27 By contrast, the Federal Circuit in *Ultramercial* found claims concerning a method for
28 distributing media products over the Internet to be patent-eligible. *Ultramercial*, 722 F.3d at 1352-

1 53. There, the claims required steps to be performed through an “extensive computer interface”
2 involving “eleven separate and specific steps with many limitations and sub-steps in each
3 category.” *Id.* There, the Federal Circuit reiterated that “[m]ere reference to a general purpose
4 computer will not save a method claim from being deemed too abstract to be patent eligible.” *Id.* at
5 1348 (citing *Bilski*, 130 S.Ct. at 3227). However, it held that the requirements of the patent claims
6 at issue in *Ultramercial* meaningfully limited the abstract formulae and algorithms at the core of the
7 claims so that the claimed invention was not “so manifestly abstract as to override the statutory
8 language of section 101.” *Id.* (quoting *Research Corp. Technologies, Inc. v. Microsoft Corp.*, 627
9 F.3d 859, 869 (Fed. Cir. 2010) [upholding claimed method for rendering a halftone image by
10 manipulation of computer data and output]).

11 It is with these principles and authorities in mind that the Court turns to the question of
12 patent eligibility presented here.

13 **II. APPLICATION TO PATENTS-IN-SUIT**

14 The Court agrees that the claims here do not state patentable subject matter. The patents
15 describe their invention as “an improved method for performing visibility calculations” in three-
16 dimensional graphics. (‘047 Patent, col. 2:23-24; ‘679 Patent (‘679 Patent), col. 2:19-20; patent
17 Titles.) Visibility calculations are used for “visible surface detection,” which is “one of the most
18 basic operations” in 3D graphics. (‘047 Patent, col. 1:18-21; ‘679 Patent, col. 1:16-18.) The
19 claimed methods of the patents shorten the calculations by *identifying* the parts of the surfaces in a
20 3-D representation that are always visible or always invisible from a set of viewpoints, and then
21 *skipping* the calculations for that part in further rendering of the image. The ‘047 and ‘679
22 preambles claim methods of reducing “visibility calculations” (or “computations”) “in 3-D
23 computer graphics” (‘047) and “required for the production of multidimensional computer
24 generated images” (‘679). As agreed by the parties, these phrases mean: “decreasing the number of
25 computational operations required to perform prior art visibility computations” “in the field of
26
27
28

1 three-dimensional computer graphics” (‘047) or “to produce a 3-D image.” (‘679). (Dkt.No. 64 at
2 5.)⁵

3 All of the Asserted Claims adhere to the same basic format as independent claim 1 of the
4 ‘047 Patent. First, the claims include a preamble that states that the method is for reducing the
5 visibility related computations calculations in a certain field of use (such as “3-D computer
6 graphics”), followed by method steps to perform the reduction. The method steps have three
7 components:

8 (1) mathematical manipulations of data, such as “identifying,” “comparing,” or
9 “determining;”

10 (2) a reducing step (“ignoring” or “exempting” or “skipping” data or calculations); and

11 (3) an indication of where these steps fit into the visibility computations.

12 Finally, there are a few references to generic computing structures (*e.g.*, “data,” a “record,” or a
13 “computer”). In sum, the patents claim a mathematical principle for eliminating or simplifying
14 portions of otherwise well-known and established computations by pre-determining a portion of the
15 calculations, and then specifying that a portion of the calculations can be skipped or omitted going
16 forward. The claims specify no physical output or programming result. Rather, in every claim, the
17 last method step recites either ignoring part of the visibility computations or completing such
18 calculations without the unnecessary steps.

19
20
21
22 ⁵ The parties have defined “visibility calculations,” in relevant part, as “rendering
23 computations that determine which pixels ... of 3D surface(s) are visible,” and thus such
24 calculations already imply the production of a 3D image. (Dkt. No. 64 at 2.) The parties also
25 agreed on the following definitions of terms:

26 “projections” and “projection planes” refer to “representations of a 3D object’s surfaces on
27 an imaginary plane”;

28 “grid cells” are an “imaginary lattice structure”;

“bounding volume” is “the smallest imaginary right quadrangular prism just enclosing the
3-D object’s surfaces”; and

“surfaces” are simply “surfaces” of 3D objects that are part of a scene.

(Dkt. No. 64 at 1-3.)

1 **A. Lack of Meaningful Limitation To A Specific Application/Inventive Concept**

2 The claims do not limit the mathematical formula to any specific use beyond the broad field
3 of invention of “computer graphics.” (‘047 Patent, col. 1:11-16; ‘679 Patent, col. 1:8-13; ‘679 and
4 ‘047 Patents, Abstract [the disclosed method operates “in 3D graphics systems”].) Simply limiting
5 the use of an abstract formula to a particular field is insufficient to avoid the prohibition on
6 patenting abstract ideas. *Bilski*, 130 S.Ct. at 3230; *Mayo*, 132 S.Ct. at 1301.

7 Similarly, the claims only state a reduction in computations, with no specific output. The
8 claims cover any application for which such computations might be used. Any equations having
9 the same “mathematical relationship of entities” fall within the invention regardless of the physical
10 properties or variables to which they are applied. (‘047 Patent, col.4:6:11; ‘679 Patent, col.4:36-
11 40.) Just as the method of converting binary numbers in *Benson* would cover all computer
12 applications of that algorithm, the claims at issue here would cover all applications of their
13 algorithm in the field of “3D computer graphics.” Indeed, the patent specification indicates that the
14 applications of the formula are numerous and would include any process that requires visible
15 surface detection, such as “radiosity calculations to compute the energy interactions between
16 surfaces,” “Hidden Surface Computations” for “computer animation, flight simulation or dynamic
17 graphics,” “Ray Tracing Computations,” “Computer Vision,” “virtual reality applications,” and the
18 “processing and display of scientific data such as energy spectra data.” (‘047 Patent, col. 1:20-22,
19 15:33-41, 15:58-16:19, 16:21-59, 18:33-46, 18:46-48, 18:56-59.)

20 **B. Conventional Post-Solution Activity**

21 Moreover, the claims do not add any steps other than conventional “post-solution” activity
22 to the abstract formula described, making them unpatentable as stated in *Flook* and *Mayo*. As the
23 patent specifications explain, “[v]isible surface detection is one of the most basic operations in 3D
24 graphics” for generating 3D images. (‘047 Patent, col. 1:18-21, 2:25-41; ‘679 Patent, col. 1:16-18,
25 2:20-27.) The patents purport to make surface detection computations faster by computing the
26 “surface elements obviously visible or invisible to each other,” and ignoring those elements,
27 thereby “reducing the visibility related computations.” (‘047 Patent, col. 1:40-43, 2:21-44; ‘679
28 Patent, col. 1:38-40, 2:18-28.) The claims confirm that conventional visibility computations follow

1 the method steps. Claim 1 of the ‘679 Patent recites that the entire method is performed “prior to
2 an occlusion or invisibility relationship computation (*known per se*) being carried out.” (‘679
3 Patent, col. 28:31 (emphasis added).) The parties agree that this refers to “prior art types of
4 rendering computations that determine which pixels or groups of pixels of 3D surface(s) are visible,
5 which include z-buffering and other visibility tests performed during rendering.” (Dkt. No. 64 at 2.)
6 The claims simply recite the steps of determining what part of well-known visibility computations
7 may be omitted, omitting them, and carrying out the remaining computations as before. They do
8 not alter the remainder of the rendering process of which those computations are a part.

9 Thus, the steps that follow are visibility computations that would be performed even if the
10 claimed method were not used. There is nothing in the patents’ claims or specifications that adds to
11 the formula steps, or a combination of steps, that would transform the otherwise abstract formula
12 into an “inventive concept.” *See Mayo*, 132 S Ct. at 1299. As in *Flook*, the claims are drawn to a
13 mathematical formula and merely append conventional activity, which cannot supply the
14 “particular application” needed to impart patentability. *Mayo*, 132 S. Ct. at 1299 (citing *Flook*).

15 C. Inclusion of General Computer Components

16 Finally, the recitation that the method is to be used on conventional computer components
17 does not make the abstract formula patentable. The parties’ constructions of the terms describing
18 the physical components on which the patent method is practiced confirm that those structures are
19 simply generic computer components. The physical components recited in the claims include
20 “computer,” “computer storage,” and “z-buffers.” “Computer” was given no construction, while
21 “computer storage” is agreed to mean simply “computer memory.” The term “z-buffer” was agreed
22 to mean “a data structure in memory that is used to store depth data,” again a generic term. (Dkt.
23 No. 64 at 6.)

24 The inclusion of these computer components does not impose a limitation sufficient to take
25 the patents out of the realm of the abstract. As noted by Defendant, five of the claims here —
26 claims 1, 4, and 5 of the ‘679 Patent and claims 1 and 12 of the ‘047 Patent— were the subject of
27 prior litigation in which they were found to fail the machine-or-transformation test. *Fuzzysharp*
28 *Techs., Inc. v. 3D Labs, Inc.*, 2009 WL 4899215, at *5 (N.D. Cal. Dec. 11, 2009). There the district

1 court, examining some of the exact claims at issue here, found that the claimed sequence of
2 “‘identifying,’ ‘comparing,’ ‘determining,’ and ‘ignoring’ data” “may be” performed on a computer
3 but does not require “any *particular* computer.” *Id.* at *4. Nor did such elements as a “method of
4 reducing the []visibility related computations in 3-D graphics,” “computer storage,” “using a data
5 structure in a computer,” and “projecting 3D images ‘on a computer screen’” “impose any
6 meaningful limit on the claim scope” because it merely “serve[d] to perform the computation.” *Id.*
7 at *5 n.3. On appeal, the Federal Circuit found:

8 [t]he references to a computer in claim 12 impose only two limitations: the
9 machine must be able to compute, and it must be able to store data. Those
10 functions are essentially synonymous with the term “computer” and thus add
11 little or nothing to simply claiming the use of a general purpose computer. The
12 recitation of computer functions in the claim thus does not confine the
preemptive effect of the claim because the underlying method has “no
substantial practical application except in connection with a digital computer.”

13 *Fuzzysharp Technologies Inc. v. 3DLabs Inc., Ltd.*, 447 F. App'x 182, 185 (Fed. Cir. 2011), reh'g
14 denied (Dec. 20, 2011) (quoting *Benson*, 409 U.S. at 71). Thus the Federal Circuit agreed with the
15 district court’s conclusion that the claims were drawn to an abstract mathematical formula. *Id.* at
16 184. However, the district court had relied exclusively on the machine-or-transformation test,
17 which the Supreme Court subsequently held was not a definitive test of patent eligibility for a
18 method claims, but simply “a useful and important clue” to patent-eligibility. *Bilski*, 130 S. Ct. at
19 3227. Thus, the Federal Circuit vacated and remanded the judgment in the *3DLabs* action to permit
20 further consideration in light of *Bilski* and to allow for claim construction related to eligibility of
21 one unspecified claim there. *Id.* at 186.

22 The Court agrees with the determination in the *3DLabs* case that the computer components
23 included in the language of the Asserted Claims does not provide a meaningful limitation on the
24 abstract mathematical formula so as to render it patent eligible. The stated components are no more
25 specific than the computer components or computer-aided processing described repeatedly in the
26 case law as insignificant to patentability. As in *Benson*, *Accenture*, and similar authorities, the
27 specification of the method as requiring use of generic computer memory does not meaningfully
28 limit the broad concept set forth in the claims. See *Benson*, 409 U.S. at 67; *Accenture*, 728 F.3d at

1 1345-46. Adding a computer storage step to an otherwise unpatentable process, even where the
2 claim names a particular type of storage component like the z-buffer here, does not transform it into
3 a claim that is patent-eligible. *Cybersource*, 654 F.3d at 1375-76 (citing *Benson*, 409 U.S. at 73-
4 74). Here, no claim recites any specific programming or computer component, but only geometric
5 calculations. Performing those calculations on a computer, even if their only practical application
6 is in a computing environment, does not render them patent-eligible. *Id.*

7 The eight additional Asserted Claims of the '047 Patent, not at issue in *3DLabs*, consist of
8 the same method steps as the others (*i.e.*, identifying, comparing and determining visibility data,
9 and then ignoring a set of data for certain surfaces in carrying out standard visibility calculations
10 thereafter). As a result, those claims likewise suffer from the same lack of meaningful limitations
11 on the mathematical formula at the core of the patent claims.

12 Unlike *Ulramercial*, the claims here suggest purely mathematical efficiencies, and do not
13 require an “extensive computer interface” involving multiple steps and limitations. *Ulramercial*,
14 722 F.3d at 1352-53. Nor do the claims in the patents-in-suit implicate particular physical elements
15 beyond a generic computer environment for carrying out their steps. *Cf. Research Corp.*
16 *Technologies*, 627 F.3d at 869 (invention describing halftone rendering process in which digital
17 data processor was utilized, along with “high contrast film,” “film printer,” as well as generic
18 computer devices such as “memory” and “printer and display devices,” was not so abstract as to be
19 lose patent eligibility).

20 **III. CONCLUSION**

21 Based upon the foregoing, Intel’s Motion for Summary Judgment for patent ineligibility of
22 the Asserted Claims under Section 101 is **GRANTED**.

23 Intel is directed to submit a proposed form of judgment forthwith.

24 This Order terminates Docket No. 28.

25 **IT IS SO ORDERED.**

26 **Date: November 7, 2013**

27 
28 **YVONNE GONZALEZ ROGERS**
UNITED STATES DISTRICT COURT JUDGE