

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTEL CORPORATION,
Petitioner,

v.

FUZZYSHARP TECHNOLOGIES, INC.,
Patent Owner.

Case IPR2014-00002
Patent 6,172,679 B1

Before JUSTIN T. ARBES, TREVOR M. JEFFERSON, and
DAVID C. MCKONE, *Administrative Patent Judges*.

MCKONE, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

A. Background

Intel Corporation (“Petitioner”) filed a Substitute Petition (Paper 4, “Pet.”) to institute an *inter partes* review of claims 1, 4, and 5 of U.S. Patent No. 6,172,679 B1 (Ex. 1001, “the ’679 patent”). FuzzySharp Technologies, Inc. (“Patent Owner”) filed a Preliminary Response (Paper 8, “Prelim. Resp.”). Pursuant to 35 U.S.C. § 314, in our Decision to Institute (Paper 9, “Dec.”), we instituted this proceeding as to all of the challenged claims of the ’679 patent.

During this trial, Patent Owner filed a Patent Owner Response (Paper 16, “PO Resp.”) and Petitioner filed a Reply to the Patent Owner Response (Paper 19, “Reply”). An oral hearing in this matter and IPR2014-00001 (argued together) was held on October 28, 2014 (Paper 22, “Tr.”).

We have jurisdiction under 35 U.S.C. § 6(c). This decision is a final written decision under 35 U.S.C. § 318(a) as to the patentability of the challenged claims. Based on the record before us, Petitioner has demonstrated by a preponderance of the evidence that claims 1, 4, and 5 are unpatentable.

B. Related Proceedings

According to Petitioner, Patent Owner has asserted the ’679 patent against Petitioner in Case No. 4:12-cv-04413-YGR (N.D. Cal.) (“Intel action”), which is currently on appeal to the U.S. Court of Appeals for the Federal Circuit. Pet. 2; Paper 18.

According to Patent Owner, the only matter pending that may be affected by a decision in this proceeding is *FuzzySharp Technologies, Inc. v. Nvidia Corp.*, Civil Action No. 12-cv-6375-JST (N.D. Cal.), filed on December 17, 2012. Paper 6, at 2.

Petitioner also filed a petition for *inter partes* review of Patent 6,618,047 B1 (“the ’047 patent”). See *Intel Corp. v. FuzzySharp Technologies, Inc.*, Case IPR2014-00001 (PTAB Sept. 30, 2013) (Paper 1). The ’047 patent also is asserted by Patent Owner in the Intel action. See, e.g., Ex. 1009.

C. Reference Relied Upon

Petitioner relies upon the following prior art reference: David Salesin & Jorge Stolfi, *The ZZ-Buffer: A Simple and Efficient Rendering Algorithm with Reliable Antialiasing* (1989) (Ex. 1002, “Salesin”).

D. Ground of Unpatentability

We instituted this proceeding based on the ground of anticipation of claims 1, 4, and 5, under 35 U.S.C. § 102(b), by Salesin.

E. The ’679 Patent

The ’679 patent describes techniques for improving three-dimensional (“3-D”) computer graphics visibility calculations. Ex. 1001, 1:9–12. A point in a 3-D image can be viewed from multiple viewpoints. *Id.* at 4:44–54. This is illustrated in Figures 2 and 3, reproduced below:

A group of viewpoints (e.g., the three VPs of Figure 3) can be associated with a coordinate system and grouped in a “viewpoint bounding box” (e.g., BB of Figure 3), which the patent describes as the smallest right quadrangular prism enclosing the viewpoints. *Id.* at 4:63–5:1. The point to be observed can be said to be totally visible from the bounding box if it is always visible from every possible viewpoint in the bounding box and totally invisible if it is hidden from every such viewpoint. *Id.* at 5:7–11. The group of viewpoints may contain only a single viewpoint; in that case, the bounding box degenerates into the viewpoint. *Id.* at 5:18–22.

As shown in Figure 2, the point to be observed (PO) can be represented by the intersection of a projection plane (PP) and a vector (VV) from the point to be observed (PO) to a viewpoint (VP). *Id.* at 5:31–43. As shown in Figure 3, to facilitate sampling, the projection plane can be divided into rectangular cells or elements (fuzzy array FA). *Id.* at 6:58–67; Fig. 2. As shown in Figure 2, the point may be included in a visible patch (PT) that occludes an invisible point (IP). *Id.* at 5:48–50.

The '679 patent describes detecting patches that are invisible to all viewpoints in a bounding box. *Id.* at 9:8–10:55. Similarly, it describes detecting patches that are totally visible to all viewpoints in the bounding box. *Id.* at 10:56–11:52. Overlapping patches can be stored in a linked list called a projection patch list. *Id.* at 11:56–61. The '679 patent describes calculating a list of the totally visible and totally invisible patches for a viewpoint group. *Id.* at 11:56–13:26. The patches identified as totally visible and totally invisible can be ignored in subsequent visibility computations. *Id.* at 13:26–35. For example, they need not be compared with other patches to determine their visibility. *Id.* at 13:31–33.

Claim 1, reproduced below, is illustrative of the claimed subject matter:

1. A method of reducing the complexity of visibility calculations required for the production of multidimensional computer generated images, said method performed on a computer, said method comprising the steps of:

prior to an occlusion or invisibility relationship computation (known per se) being carried out on a plurality of surfaces from each viewpoint to be calculated:

for selected ones of said surfaces, 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;

wherein said remaining surface is a surface which is unable to be determined with certainty as to whether it is either unoccluded or hidden;

exempting from said occlusion or invisibility relationship computation those surfaces which are either always unoccluded or always hidden;

maintaining a record of said remaining surface; and

carrying out occlusion or invisibility relationship computations on said remaining surfaces.

II. ANALYSIS

A. Claim Construction

The '679 patent is expired. “[T]he Board’s review of the claims of an expired patent is similar to that of a district court’s review.” *In re Rambus Inc.*, 694 F.3d 42, 46 (Fed. Cir. 2012) (internal citations omitted). Thus, we construe the claims in accordance with their ordinary and customary meanings, as would be understood by a person of ordinary skill in the art, in the context of the specification. *See generally Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

In the Petition, Petitioner asserted that the parties had stipulated to the constructions of several claim terms in the Intel action. Pet. 11–12, 16–18. Patent Owner did not propose a construction for any claim term in the Preliminary Response. For these claim terms, we preliminarily determined that the purportedly stipulated constructions proposed by Petitioner represented the ordinary and customary meaning of those terms, as would be understood by a person of ordinary skill in the art, in the context of the specification. Dec. 9–10.

Additionally, Petitioner proposed a construction of “determining for said viewpoint whether each . . . 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,” as recited in claim 1. Petitioner contended that, in the Intel action, the parties disputed “whether this step is satisfied by determining both which surfaces are invisible and which surfaces are visible (Intel’s position) or whether it requires merely making one or the other of those determinations (the Patent Owner’s position).” Pet. 13. We

preliminarily determined that Salesin anticipates claims 1, 4, and 5 under either interpretation and, thus, declined to construe the term expressly.

Dec. 10–11.

In sum, we preliminarily construed the claim terms of the '679 patent as follows:

Claim Term	Construction
“visibility calculations” (claims 1, 4, 5) “occlusion or invisibility relationship computation (known per se)” (claims 1, 4, 5)	prior art types of rendering computations that determine which pixels or groups of pixels of 3-D surface(s) are visible, which include z-buffering and other visibility tests performed during rendering
“reducing the complexity of visibility calculations” (claims 1, 4, 5) “reduction in the number and/or duration of visibility calculations” (claim 5)	decreasing the number of computational operations to perform visibility computations
“surfaces” (claims 1, 4, 5)	surfaces (i.e., the exterior) of 3-D objects that are part of a scene
“viewpoint” (claims 1, 4, 5)	the spatial position and orientation of a viewer
“always unoccluded” (claims 1, 4, 5) “always hidden” (claims 1, 4, 5)	totally invisible/visible from a single viewpoint (if a single viewpoint) or totally invisible/visible from every viewpoint (if from a group of viewpoints)
“graphic images” (claim 4)	an image
“computer vision data” (claim 4)	data representing images
“abstract data” (claim 4)	data derived from physical data

Claim Term	Construction
“determining for said viewpoint whether each . . . 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” (claims 1, 4, 5)	(does not require express construction)

Petitioner and Patent Owner do not dispute these constructions. We see no reason to change these constructions based on the complete record developed during this trial and, thus, maintain them for purposes of this Decision. Neither Petitioner nor Patent Owner identifies any additional terms requiring construction. We also do not identify any other terms that require express construction for purposes of this Decision.

B. Asserted Grounds of Unpatentability

1. Anticipation of Claims 1, 4, and 5 by Salesin

a. Salesin

Salesin describes an improved algorithm for rendering 3-D images. Ex. 1002, p. 1, cols. 1–2.¹ According to this algorithm, a screen plane is divided into a two-dimensional array of rectangular cells, each of which covers a fixed-sized block of pixels. *Id.* at p. 2, col. 1. This array is represented in memory by a “ZZ-buffer,” which is a two-dimensional array. *Id.* Figure 1, reproduced below, illustrates an example:

¹ Citations to Exhibit 1002 are in the form of page number of the article and column number on the page (p. x, col. y).

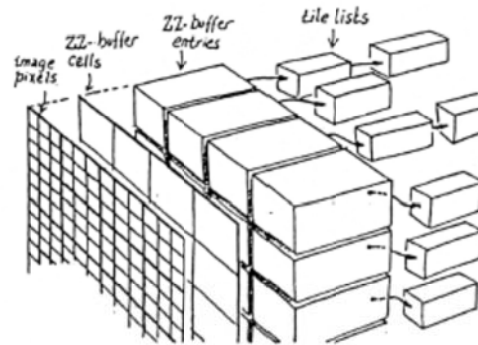


Figure 1. The ZZ-buffer.

Figure 1 illustrates a ZZ-buffer. Each ZZ-buffer entry includes a pointer to a “tilelist,” or linked list of tiles, and each tile describes an object that may be visible within the ZZ-buffer cell. *Id.* at p. 2, col. 2. Figure 2, reproduced below, illustrates Salesin’s tiles:

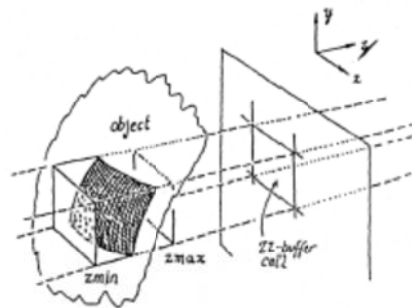


Figure 2. Tiles.

Figure 2 illustrates the representation of a tile in a tilelist. Each tile includes minimum and maximum depths (Z_{min} and Z_{max} , respectively, in the “z” direction of a coordinate system) spanned by the object in the tile relative to the screen plane. *Id.* Each tile also has a field indicating whether its object is opaque and covers the cell entirely. *Id.*

During a “scan conversion” phase, tiles can be added to the tilelists of the cells. *Id.* at p. 3, col. 1. During the tile adding procedure, a quick visibility test is performed on a new tile based on its depth and opacity information, and the tile is thrown away if it is obscured completely by other tiles in the tilelist. *Id.* at p. 2, col. 2–p. 3, col. 1. Similarly, if the new tile completely obscures other tiles in the tilelist, those other tiles in the tilelist are discarded. *Id.* at p. 3, col. 1. Salesin describes a “rendering” phase, performed after the scan conversion phase, during which the visibility of objects is further calculated and visible objects are shaded. *Id.* at p. 3, col. 1–2. According to Salesin, if the first object in a tilelist is opaque and covers the entire cell, then no visibility tests are performed for that cell. *Id.* at p. 12, cols. 1–2.

b. Claim 1

We have considered the evidence and argument presented in the Petition (at 20–30), including the testimony of Dr. Seth Teller (Ex. 1013, “Teller Decl.”)², a professor in the Electrical Engineering and Computer Science department of the Massachusetts Institute of Technology; the PO Response; and Petitioner’s Reply, and conclude that Salesin discloses each limitation of claims 1, 4, and 5.

We find that Salesin discloses a “method of reducing the complexity of visibility calculations required for the production of multidimensional computer generated images, said method performed on a computer,” as

² The Petition cites to Exhibit 1008. Petitioner, however, filed Exhibit 1013 correcting Exhibit 1008. *See* Paper 11, at 2–3; Paper 13, at 1–2.

recited in claim 1. Specifically, Salesin describes a two-phase process for generating 3-D images: scan conversion, during which a ZZ-buffer is created, and rendering, where visibility ultimately is determined. Ex. 1002, p. 3, cols. 1–2, p. 5, col. 1. During the rendering phase, the visibility of an object in the ZZ-buffer is further refined “if it has not already been resolved by the ZZ-buffer.” *Id.* at p. 3, cols. 1–2. Salesin’s scan conversion step reduces the number of visibility calculations performed during the rendering phase. Ex. 1013 ¶ 90.

Salesin’s “scan conversion” phase is performed prior to the “rendering” phase, in which “the visibility of the objects within each pixel is further refined (if it has not already been resolved by the ZZ buffer), visible objects are shaded, and the shades are blended to produce the final pixel colors.” Ex. 1002, p. 3, cols. 1–2. The rendering phase relies on the depth and opacity information determined in the scan conversion phase. Ex. 1002, p. 5, col. 1. Thus, Salesin’s scan conversion phase is performed “prior to an occlusion or invisibility relationship computation (known per se) being carried out on a plurality of surfaces from each viewpoint to be calculated,” as recited in claim 1 and construed in Section II.A, above.

We also find that Salesin discloses:

for selected ones of said surfaces, 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.

Specifically, Salesin discloses that, in its “scan conversion” phase, tiles (stored objects to be rendered) are added to and removed from tilelists. Ex. 1002, p. 2, col. 2–p. 3, col. 1. When a new tile is added to a tilelist for a cell, the new tile is compared to the existing tiles to determine whether it is obscured completely or whether it completely obscures other tiles. *Id.* at p. 2, col. 2–p. 3, col. 1. The new tile is thrown away if it is completely obscured by other tiles in the tilelist. *Id.* This is an example of determining that the tile is an always hidden surface. Ex. 1013 ¶¶ 91–92. If the new tile is opaque and completely obscures the tiles already in the list, the tiles already in the list are thrown away. Ex. 1002, p. 3, col. 1. This is an example of determining that the tile is an always unoccluded surface. Ex. 1013 ¶¶ 91–92.

Salesin also discloses determining for a viewpoint whether a surface is a remaining surface. Claim 1 recites that a “remaining surface is a surface which is unable to be determined with certainty as to whether it is either unoccluded or hidden.” According to Salesin, during the scan conversion phase, if a tile is not determined to be obscured completely or to obscure completely the tiles already in the list, “then the new object is appended to the list of potentially visible objects.” Ex. 1002, p. 3, col. 1. The visibility or invisibility of potentially visible objects is determined later in the rendering phase. *Id.* at p. 3, cols. 1–2.

Patent Owner argues that Salesin’s “quick visibility test” during the scan conversion phase does not result in a tile being placed into one of the categories “always unoccluded,” “always hidden,” or “remaining.” PO Resp. 2. Rather, Patent Owner argues, Salesin discloses a new category, “potentially visible object.” *Id.* Patent Owner contends that Salesin does

not disclose a “remain[ing] surface” because Salesin’s technique keeps testing surfaces, sometimes repeatedly, to determine visibility. *Id.*

We are not persuaded by Patent Owner’s arguments. Claim 1 recites that a “remaining surface” is “a surface which is unable to be determined with certainty as to whether it is either unoccluded or hidden.” Salesin describes a “potentially visible object” as a tile that is not determined to be obscured completely by other tiles (hidden) or that is not determined to obscure completely the other tiles in a list (unoccluded). Ex. 1002, p. 2, col. 2–p. 3, col. 1. Patent Owner appears to be arguing that a tile, during one iteration of Salesin’s technique, may be determined to be unoccluded and, yet, in a later iteration, may be determined to be occluded, and, thus, the tile is not put definitively into any category. PO Resp. 2. Salesin, however, is clear that, if the top tile in a tilelist is opaque and completely occludes the tiles below it, “no visibility tests are performed” during the rendering phase. Ex. 1002, p. 12, cols. 1–2. Thus, the tile is categorized definitively. Moreover, nothing in Salesin suggests that a tile deemed to be occluded (and thrown away) may later be unoccluded. *Id.* at p. 12, col. 1. Thus, Salesin discloses definitively categorizing tiles as unoccluded or hidden. Ex. 1013 ¶ 91.

We also discern no difference between Salesin’s “potentially visible object[]” and claim 1’s “remaining surface.” In both cases, a surface that cannot be categorized definitively into one of the other categories (unoccluded or hidden) is put into a category of surfaces that must be dealt with in more detail during a rendering phase. Ex. 1013 ¶¶ 92–93.

In sum, we find that this set of limitations is disclosed in Salesin. We note that this is the case regardless of whether this set of limitations requires

determining both which surfaces are invisible and which surfaces are visible or whether it requires merely making one or the other of those determinations.

We also find that Salesin discloses “exempting from said occlusion or invisibility relationship computation those surfaces which are either always unoccluded or always hidden,” as recited in claim 1. If a tile is obscured completely (always hidden), it is discarded. Ex. 1002, p. 2, col. 2–p. 3, col. 1. If the tile is the first in a tilelist and completely obscures the remaining tiles (is always unoccluded), no visibility tests are performed on the cell. Ex. 1002, p. 12, cols. 1–2. Thus, no visibility computations are performed in Salesin’s rendering phase on tiles discarded during the scan conversion phase. Ex. 1013 ¶ 93.

At the oral hearing, Patent Owner argued that Salesin is inconsistent on whether a tile is thrown away if it is obscured by other tiles in the tilelist. Tr. 32:16–34:3. In particular, Salesin states that “the tiles already in the list are thrown away if they are completely obscured by the new one” and later states:

Note that this operation takes only a constant amount of time per object and per cell, because whenever the z interval of the new tile overlaps that of the entire list, we just append the new tile to the list. We do not attempt to throw away the individual tiles that are obscured by the new tile because the time spent in this search is likely to be wasted when a subsequent tile obscures them all.

Ex. 1002, p. 3, col. 1. Patent Owner did not state at the hearing which claim limitation it contends is missing from Salesin due to this purported discrepancy. Tr. 33:5–17. Patent Owner appears not to contend that this argument applies to the ’679 patent. *Id.* at 34:4–10. In any case, claim 1

does not recite throwing away a surface determined to be hidden. Rather, it recites categorizing an object as hidden and exempting the surface from later occlusion or invisibility computations. For the reasons given above, we find that Salesin discloses these limitations regardless of whether (or when) hidden tiles are thrown away.

Because the tiles for the remaining surfaces are added to, or kept in, the tilelists, Ex. 1002, p. 2, col. 2–p. 3, col. 1, Salesin discloses “maintaining a record of said remaining surface,” as recited in claim 1.

Finally, we find that Salesin discloses “carrying out occlusion or invisibility relationship computations on said remaining surfaces,” as recited in claim 1 and construed in Section II.A, above. During Salesin’s rendering phase, “the visibility of objects within each pixel is further refined (if it has not already been resolved by the ZZ-buffer), visible objects are shaded, and the shades are blended to produce the final pixel colors.” Ex. 1002, p. 3, cols. 1–2.

Upon consideration of Petitioner’s evidence and arguments (including the testimony of Dr. Teller) and Patent Owner’s arguments, we conclude that Petitioner has proved by a preponderance of the evidence that claim 1 is anticipated by Salesin.

c. Claim 4

Claim 4 depends from claim 1. Petitioner contends that Salesin anticipates claim 4. Pet. 25, 30. Patent Owner does not introduce evidence or argument disputing Petitioner’s showing that Salesin discloses the additional limitation of claim 4. Upon consideration of Petitioner’s evidence and argument, we find that Salesin discloses “wherein said images are

selected from the group consisting of graphic images, computer vision data, abstract data and physical data,” as recited in claim 4. As explained in Section II.A, a “graphic image[]” is an image. Salesin describes a technique for rendering “high-quality images.” Ex. 1002, p. 1, col. 1.

Accordingly, we conclude that Petitioner has proved by a preponderance of the evidence that claim 4 is anticipated by Salesin.

d. Claim 5

Claim 5 depends from claim 1. Patent Owner does not introduce evidence or argument disputing Petitioner’s showing that Salesin discloses the additional limitation of claim 5. Upon consideration of Petitioner’s evidence and argument (Pet. 25, 31), we find that Salesin discloses “wherein the reduction in complexity involves a reduction in the number and/or duration of visibility calculations,” as recited in claim 5. Because the visibility of an object in Salesin’s ZZ-buffer will be refined during the rendering phase only if the visibility of the object has not been resolved during the scan conversion phase, the number and duration of visibility calculations is reduced for Salesin’s rendering phase. Ex. 1002, p. 3, cols. 1–2.

Accordingly, we conclude that Petitioner has proved by a preponderance of the evidence that claim 5 is anticipated by Salesin.

III. CONCLUSION

Petitioner has demonstrated by a preponderance of the evidence that claims 1, 4, and 5 of the '679 patent are unpatentable as anticipated under 35 U.S.C. § 102(b) by Salesin.

IV. ORDER

For the reasons given, it is

ORDERED that, based on a preponderance of the evidence, claims 1, 4, and 5 of U.S. Patent No. 6,172,679 B1 are held unpatentable; and

FURTHER ORDERED that, because this is a final written decision, parties to this proceeding seeking judicial review of our Decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

James F. Valentine
Rajiv P. Sarathy
PERKINS COIE LLP
3150 Porter Drive
Palo Alto, CA 94304
JValentine@perkinscoie.com
RSarathy@perkinscoie.com

PATENT OWNER:

David Fink
Timothy Johnson
FINK & JOHNSON
7519 Apache Plume
Houston, TX 77071
texascowboy6@gmail.com
federallitigationlaw@gmail.com