No. 20-2257

IN THE UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

NATURE SIMULATION SYSTEMS INC.,

Plaintiff-Appellant,

v.

AUTODESK, INC.,

Defendant-Appellee.

Appeal from the United States District Court
for the Northern District of California, Case No. 3:19-cv-03192-SK.
The Honorable Sallie Kim, Magistrate Judge Presiding.

JOINT APPENDIX

Matthew M. Wawrzyn
WAWRZYN LLC
200 East Randolph Street, Suite 5100
Chicago, IL 60601
(312) 235-3120
matt@wawrzynlaw.com

Counsel for Plaintiff-Appellant
Nature Simulation Systems Inc.

Appendix Table of Contents

Nature Simulation Systems, Inc. v. Autodesk, Case No. 20-2257 (Fed Cir.)

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

NATURE SIMULATION SYSTEMS INC., Plaintiff, v. AUTODESK, INC., Defendant.	Case No. 19-cv-03192-SK **ORDER REGARDING CLAIM CONSTRUCTION**

United States District Court
Northern District of California

On June 7, 2019, Plaintiff Nature Simulation Systems, Inc. ("NSS") filed a complaint alleging that Defendant Autodesk, Inc. ("Autodesk") infringes two of its patents, U.S. Patent No. 10,120,961 (the "'961 patent") and U.S. Patent No. 10,109,105 (the "'105 patent"). (Dkt. 1.) Both parties have consented to the jurisdiction of the undersigned pursuant to 28 U.S.C. § 636(c). (Dkts. 11, 13.) The Court entered a stipulated protective order on March 18, 2020. (Dkt. 32.) NSS filed its opening claim construction brief on April 27, 2020. (Dkt. 35.) Autodesk filed its brief in opposition on May 11, 2020. (Dkt. 36.) NSS replied on May 18, 2020. (Dkt. 37.) The Court held a tutorial hearing via Zoom on May 26, 2020. (Dkt. 43.) Autodesk's expert Daniel Aliaga, Ph.D., appeared and testified at the tutorial hearing. (*Id.*) NSS presented no expert witness testimony at the tutorial hearing. (*Id.*)

On June 4, 2020, the Court held a claim construction hearing via Zoom. (Dkt. 49.) The parties dispute the construction of eight terms contained in the two patents at issue. (Dkt. 48-1.) At the claim construction hearing, the Court heard oral argument regarding the first two disputed claim terms and took construction of those terms under submission, with further hearing on the remaining terms to be set if necessary. (*Id.*) Having reviewed the submissions of the parties, the relevant legal authority, and the record in the case, and having had the benefit of expert testimony and oral argument, the Court HEREBY CONCLUDES that the first two disputed terms are indefinite, rendering the patents-in-suit invalid.

BACKGROUND

The patents in dispute here disclose methods of packing data for computer aided design ("CAD"). In essence, like all CAD, the asserted patents allegedly infringed by Autodesk's software program Autodesk 3ds Max create a representation of a three-dimensional object or objects. Specifically, the patents "relate to computer-implemented methods for performing Boolean operations on three-dimensional, geometric objects." (Dkt. 36.) The '961 patent is a continuation-in-part of the application that led to the '105 patent. NSS asserts claim 1 of the '105 patent and claims 1 and 8 of the '961 patents against Autodesk. The claims are very similar.

NSS concedes that the general idea of performing Boolean operations in this area was well known before the patents in dispute, as the asserted patents cite to prior art disclosing this concept.

NSS describes the asserted patents as "strik[ing] a balance" between constructive solid geometry and boundary representation, in which the asserted patents use some of the best features of both methods of packaging data. (Dkt. 35.) NSS describes that: (1) "each three-dimensional object exists in a virtual box," (2) this "object bounded in the box is triangulated, meaning that the object is decomposed into more triangles," and (3) a "triangle in a three-dimensional space has three neighboring triangles." (*Id.*)

There are two methods from prior art that are cited in the asserted patents. The Delaunay method is a known method of triangulation (known as "Delaunay triangulation"), and the Watson method is a known algorithm for computing a Delaunay triangulation that is described in a paper written in 1981 by D.F. Watson. ('961 patent at 6:64-66, '105 patent, 6:42-44.)

DISCUSSION

A. Legal Standards.

The Constitution provides that "[t]he Congress shall have Power… To promote the Progress of Science and the useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries." Art. 1, § 8, cl. 8. Pursuant to this provision, the United States issues patents protecting "any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof." 35 U.S.C. §

101. To safeguard its subject matter, "a patent must describe the exact scope of an invention and its manufacture." *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 373 (1996). A patent's exacting description must contain two parts: a specification describing the invention "in such full, clear, concise, and exact terms as to enable any person skilled in the art … to make and use the same" and "one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention." 35 U.S.C. § 112. The claim "functions to forbid not only exact copies of an invention" but also competing "products that go to 'the heart of an invention but avoi[d] the literal language of the claim by making a noncritical change.'" *Markman*, 517 U.S. at 373-74 (citation omitted).

The owner of a patent may seek to enforce it by alleging infringement, *i.e.*, that the defendant "without authority makes, uses, offers to sell, or sells any patented invention, within the United States or imports into the United States any patented invention during the term of the patent therefor." 35 U.S.C. § 271(a). "Victory in in infringement suit requires a finding that the patent claim 'covers the alleged infringer's product or process,' which in turn necessitates a determination of 'what the words in the claim mean.'" *Markman*, 517 U.S. at 374 (citing H. Schwartz, Patent Law and Practice 1, 80 (2d ed. 1995)). Indeed, "[i]t is a bedrock principle of patent law that the claims of a patent define the invention to which the patentee is entitled the right to exclude." *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004) (collecting cases). Infringement actions therefore "requir[e] a two-step analysis;" first, construction of the claims, which is a matter of law, and second, determination of infringement, which is a matter of fact. *Wright Med. Tech., Inc. v. Osteonics Corp.*, 122 F.3d 1440, 1443 (Fed. Cir. 1997).

1. Claim Construction.

"The construction of a patent, including terms of art within its claim, is exclusively within the province of the court" as a matter of law. *Markman*, 517 U.S. at 372. "It is well-settled that, in interpreting an asserted claim, the court should look first to the intrinsic evidence of record, *i.e.*, the patent itself, including the claims, the specification and, if in evidence, the prosecution history. *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996). "Such intrinsic evidence is the most significant source of the legally operative meaning of the disputed claim language." *Id.* Because of the "bedrock principle" that the claims define the invention subject to the patent, "a claim construction analysis must begin and remain centered on the claim language itself, for that is the language the patentee has chosen to 'particularly poin[t] out and distinctly clai[m] the subject matter which the patentee regards as his invention.'" *Innova/Pure Water*, 381 F.3d at 1116 (citing *Interactive Gift Express, Inc. v. Compuserve, Inc.*, 256 F.3d 1323, 1331 (Fed. Cir. 2001) (quoting 35 U.S.C. § 112)).

Claim construction analysis proceeds in three steps. First, courts "look to the words of the claims themselves, both asserted and nonasserted, to define the scope of the patented invention." *Vitronics*, 90 F.3d at 1582. "[W]ords in a claim are generally given their ordinary and customary meaning." *Id.* "[T]he ordinary and customary meaning of a claim term is the meaning that the term would have to a person or ordinary skill in the art in question at the time of the invention, *i.e.*, as of the effective filing date of the patent application." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005). As a second step, "it is always necessary to review the specification to determine whether the inventor has used any terms in a manner inconsistent with their ordinary meaning." *Vitronics*, 90 F.3d at 1582. Claims are to be read in light of the specification, which "acts as a dictionary when it expressly defines terms used in the claims or when it defines terms by implication." *Id.* Because of this imbrication of claims and specification, "the specification is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term." *Id.* Third and finally, "the court may also consider the prosecution history of the patent, if in evidence." *Id.*

Typically, analysis of these three sources of intrinsic evidence is sufficient to clarify "any ambiguity in a disputed claim term." *Id.* at 1583. Extrinsic evidence is customarily considered only if necessary to assist in evaluating the meaning or scope of technical terms in the claims. *Pall Corp. v. Micron Separations, Inc.*, 66 F.3d 1211, 1216 (Fed. Cir. 1995). "[E]xtrinsic evidence in the form of expert testimony can be useful to a court for a variety of purposes, such as to provide background on the technology at issue, to explain how an invention works, to ensure that the court's understanding of the technical aspects of the patent is consistent with that of a person of skill in the art, or to establish that a particular term in the patent or the prior art has a particular meaning in the pertinent field." *Phillips*, 415 F.3d at 1318. Simultaneously, a court need not credit an expert's "conclusory, unsupported assertions" or testimony that conflicts with the intrinsic evidence. *Id.* When "considered in the context of the intrinsic evidence," "extrinsic evidence may be useful to the court" because it "can help educate the court regarding the field of the invention and can help the court determine what a person of ordinary skill in the art would understand claim terms to mean." *Id.* at 1319.

2. Indefiniteness.

Section 112's provision that the specification must conclude with "one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention" is also known as the definiteness requirement. *Acacia Media Techs. Corp. v. New Destiny Internet Group*, 405 F. Supp. 2d 1127, 1131 (N.D. Cal. 2005). Patents must be definite "for the protection of the patentee, the encouragement of the inventive genius of others, and the assurance that the subject of the patent will be dedicated ultimately to the public." *Gen. Elec. Co. v. Wabash Appliance Corp.*, 304 U.S. 364, 369 (1938). To that end, "[t]he inventor must 'inform the public during the life of the patent of the limits of the monopoly asserted, so that it may be known which features may be safely used or manufactured and which may not;'" crucially, "[i]n a limited field the variant must be clearly defined." *Id.* Accordingly, "[a] patent claim which fails to meet the definiteness requirement is invalid." *Acacia*, 405 F. Supp. 2d at 1132 (citing *Wabash*, 304 U.S. at 364). Like the claim construction of which it forms a part, the court's finding that a patent claim is invalid for failure to meet the definiteness requirement is a question of law. *Bancorp Servs., LLC v. Hartford Life Ins. Co.*, 359 F.3d 1367, 1371 (Fed. Cir. 2004).

Once issued, a patent is presumed to be valid. 35 U.S.C. § 282. When a party alleges that a patent is invalid as indefinite, the court must make a determination of definiteness based on the understanding of a person of ordinary skill in the art at issue. To determine whether a claim is definite, a court must analyze whether a person ordinarily skilled in the art would understand what is claimed when the claim is read in light of the specification. *Personalized Media*

Case 3:19-cv-05792-SK Document 61 Filed 07/31/20 Page 6 of 19

Communications, LLC v. Int'l Trade Comm'n, 161 F.3d 696, 705 (Fed. Cir. 1998). "[A] claim is not indefinite merely because it poses a difficult issue of claim construction; if the claim is subject to construction, *i.e.*, it is not insolubly ambiguous, it is not invalid for indefiniteness." *Bancorp*, 359 F.3d at 1371. Courts must find "claims indefinite only if reasonable efforts at claim construction prove futile," and close questions of indefiniteness must be resolved in favor of the patentee. *Id.*

A court employs the canons of claim construction to determine definiteness. *Oakley, Inc. v. Sunglass Hut Int'l*, 316 F.3d 1331, 1340-41 (Fed. Cir. 2003). As discussed above, the three primary sources of intrinsic evidence are consulted, followed by extrinsic evidence if necessary. *Vitronics*, 90 F.3d at 1582; *Pall*, 66 F.3d at 1216. Terms are given their ordinary meaning, and technical terms included in a patent claim should be given the meaning a person experienced in the field of the invention would give them. *Verve, LLC v. Crane Cams, Inc.*, 311 F.3d 1116, 1119 (Fed. Cir. 2002) ("While reference to intrinsic evidence is primary in interpreting claims, the criterion is the meaning of words as they would be understood by persons in the field of the invention."). "Testimony by a witness, who is recognized by the Court as an expert in the field of the invention, about the common meaning of a technical term at the time the application was filed, is instructive in ascertaining its meaning" for purposes of a definiteness analysis. *Acacia*, 405 F. Supp. 2d at 1132 (citing *Glaxo Wellcome, Inc. v. Andrx Pharm., Inc.*, 344 F.3d 1226, 1229 (Fed. Cir. 2003)).

If there is an amendment to a patent, as there is here, by the United States Patent and Trademark Office ("PTO"), the PTO's amendment is presumed valid, and the party challenging the patent must present clear and convincing evidence to overcome the presumption. *Microsoft Corp. v. i4i Ltd. P'ship*, 564 U.S. 91, 95-96 (2011) (PTO must make various factual determinations" which can be set aside only for clear and convincing evidence). However, even if there is an amendment, the claim may still be found to be indefinite. *See, e.g.*, *Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1321-23 (Fed. Cir. 2012); *Sigram Schindler Beteiligungsgesellschaft mbH v. Cisco Sys., Inc.*, 726 F. Supp. 2d 396, 424-26 (D. Del. 2010).

/ / /

United States District Court
Northern District of California

B. Analysis.

Although the parties dispute seven claims in the asserted patents (Dkts. 35, 36), the Court considered the first two disputed claims, which are dispositive. The disputed claim language from both asserted patents are as follows: (1) "modified Watson method" and (2) "search neighboring triangles of the last triangle pair that holds the last intersection point."

1. "modified Watson method."

NSS proposes that the first claim ("modified Watson method") be given its "ordinary meaning." Autodesk argues that it is indefinite. The Watson method is an algorithm for computing a Delaunay triangulation that is described in a 1981 paper by D.F. Watson cited in the asserted patents. ('961 patent at 6:64-66; '105 patent, 6:42-44.) NSS and Autodesk seemingly agree on a common understanding of the Watson method. The distinction arises in the manner in which the asserted patents attempt to modify the Watson method.

To support its argument that "modified Watson method" is ambiguous, Autodesk provides the declaration of expert witness Daniel Aliaga, Ph.D. ("Aliaga"), who opined that the term "modified Watson method" does not have and did not have at the time the patents were issued an ordinary and customary meaning to a person of ordinary skill in the art ("POSITA"). (Dkt. 36-1.) Aliaga opines that a POSITA is a person with at least a master's degree in computer science or a related field, or a bachelor's degree in computer science or a related filed plus two years of relevant experience, with experience in computer graphics, computer-aided design, solid modeling, or geometric modeling. (Dkt. 36-1 ¶ 14.)

The PTO rejected the '961 patent partially because the term "modified Watson method" was not adequately defined. (Dkt. 37-1 (Ex. A (File History) at 1).) Later, the PTO amended the '961 patent to add language regarding removing duplicate intersection points, identifying positions of end intersection points, and splitting portions of each triangle. (*Id.* at 2-3.)

NSS argues, with no citation to an expert or any other evidence, that a POSITA would understand the limitations of the claim based on the intrinsic evidence of the patents. However, in the face of the undisputed opinion evidence by Aliaga, the Court cannot simply accept NSS's unsupported opinion.

Thus, the question is thus: if the PTO issues a patent after amendment to clarify an indefinite term, but an expert later opines that a POSITA would not understand the term, how does the Court determine whether the term is indefinite? The only way to do so here is to look at each argument to see if Autodesk raises any unanswered questions. Here, Autodesk does.

NSS does not provide any expert testimony to rebut Aliaga's opinion. NSS instead cites to the intrinsic evidence of the patents, specifically Fig. 13, reproduced below.

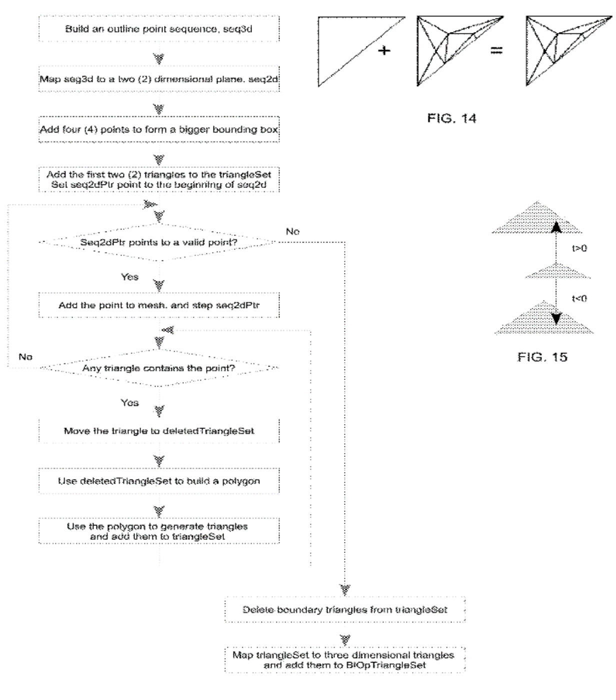


FIG. 13 Prior Art except the first two (2) steps , the last one, and the condition Any triangle contains the point.

The language of the asserted patents describes the modified Watson method as including "removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion and a middle portion." ('961 patent at 9:35-40; '105 patent, 8:64-9:3.)

i. "removing duplicate intersection points."

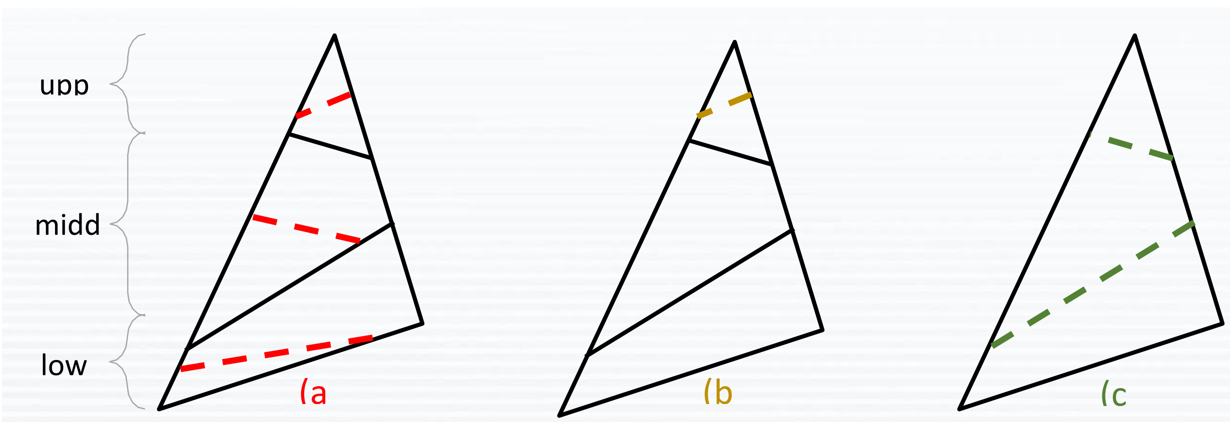
Autodesk claims that "removing duplicate intersection points" is ambiguous because the patent does not provide information telling a POSITA how to identify or how to remove duplicate intersection points. NSS points to specific language in the patents that teaches how "an intersection line passes through a set of triangles and divides each triangle into multiple

United States District Court
Northern District of California

partitions," with deletion of duplicate points. ('961 patent at 6:28-58.) The claim language leaves unanswered the following questions: (1) What is a neighboring point of intersection (referred to as "PET" in the patents)? ; (2) What is the meaning of an "identical" point of intersection? ; (3) What is the meaning of removing a point of intersection? ; (4) From what is the point of intersection being removed? NSS does not respond to these specific questions, which highlight the ambiguity of the claim language.

ii. "splitting portion of each triangle including an upper portion, a lower portion and middle portion."

Autodesk argues that this language is ambiguous because there are several ways to split a triangle into an upper, lower, and middle portion and provides a visual example, reproduced below.



(Dkt. 36-1 (Aliaga Dec. ¶ 16) (explanation); Dkt. 51 (illustration).) NSS suggests in its reply brief that it is not necessary to split the triangle into an upper, lower, and middle portion. (Dkt. 37 at page 6.) However, the claim language includes the word: "and." ('961 patent at 9:35-40.) In other words, the splitting of the triangle into the upper portion, middle portion, and lower portion is not optional, based on the language of the claim. In support of its assertion that splitting into the three portions is optional, NSS points to the specification's discussion of optional splitting of partitions. (Dkt. 37 at page 6.) This argument does not enable NSS to succeed in re-writing the language of the claim itself.

NSS points to Figure 13, *supra*, as the explanation of the modified Watson method. However, the language of Figure 13 does not match the claim language, as it does not describe in

United States District Court
Northern District of California

any manner the removing of duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion and a middle portion.

NSS points to column 7 of the specification and relies on step 5(a) of column 7 to show the modification of the Watson method. ('961 patent at 7:17-23). This language adds an additional condition: "or last segment passes through the triangle." Again, this language is not contained in the claim language and does not explain the challenged claim language.

Autodesk points to a comparison between that language of column 7: 17-23 with dependent claim 6. Column 7: 17-23 states: "For every point, check every triangle whether its circumcircle contains the point or the last segment passes through the triangle. (Dkt. No. 31-2 ('961 Patent), col. 7:19-23.) Dependent claim 6 states: "checking every triangle in the triangle set whether its circumcircle contains the point or the last segment passes through the triangle[.]" Under the doctrine of claim differentiation, the presence of a limitation in a dependent claim gives rise to the proposition that it is not the independent claim. *Halliburton Energy Servs., Inc. v. M-I LLC*, 514 F.3d 1244, 1251 n.3 (Fed. Cir. 2008). Thus, the references in dependent claim 6 do not define the term "modified Watson claim" in the manner that NSS suggests.

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

Finally, the comparison of Figure 13, *supra*, with Figure 12, reproduced below, does not lend clarity.

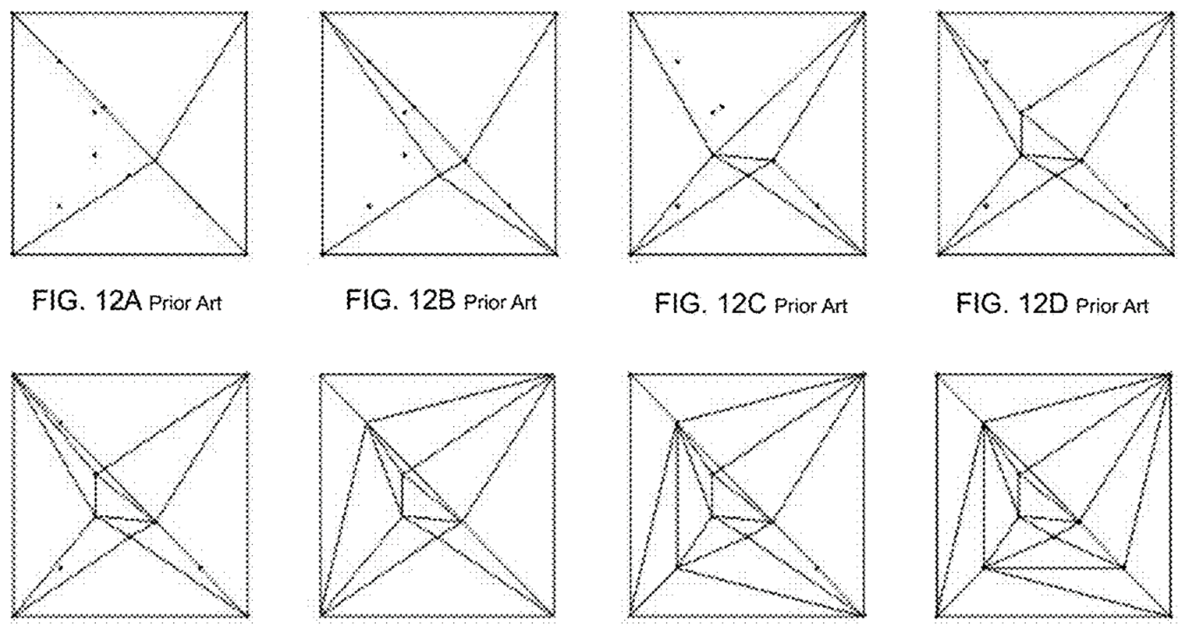


FIG. 12A Prior Art FIG. 12B Prior Art FIG. 12C Prior Art FIG. 12D Prior Art

FIG. 12E Prior Art FIG. 12F Prior Art FIG. 12G FIG. 12H

Figure 13, the flowchart, results in Figure 12. Figures 12A through 12F are labeled as "prior art" in the '961 patent but not in the '105 patent. However, if one assumes that 12G shows the "modified Watson method," as in the progression from prior art of Figure 12F to the modified Watson method of Figure 12G, an examination shows that Figure 12 does not follow the conditions of the modified Watson method as described in the patents. First, the progression from Figure 12F to Figure 12G shows that a polygon, not a triangle, is split, as demonstrated by the following depictions in color:

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

/ / /

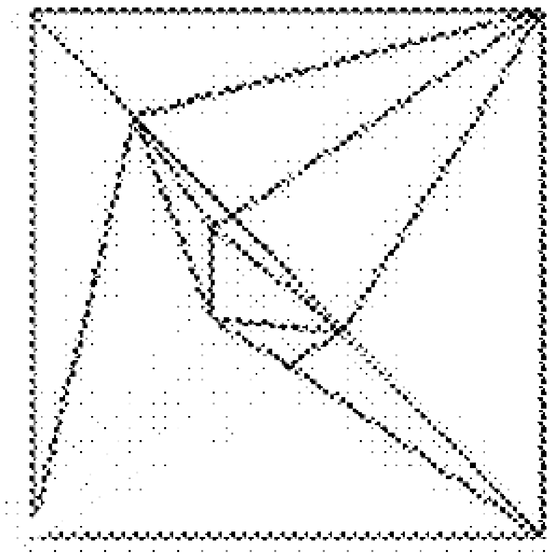


FIG. 12F Prior Art

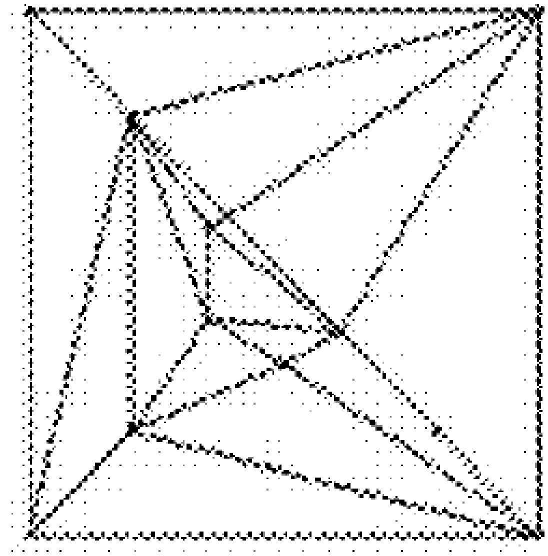


FIG. 12G

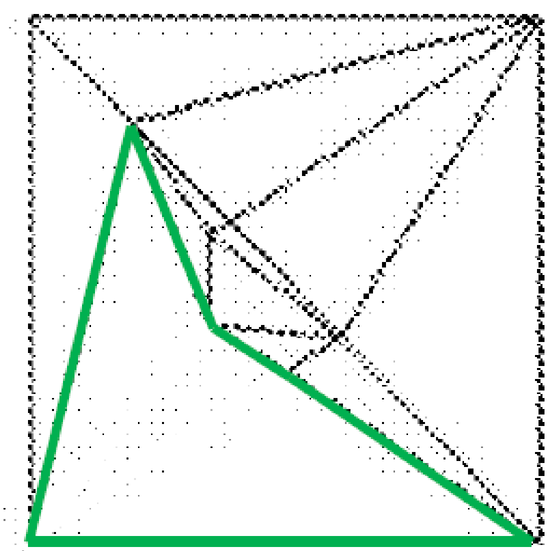


FIG. 12F Prior Art

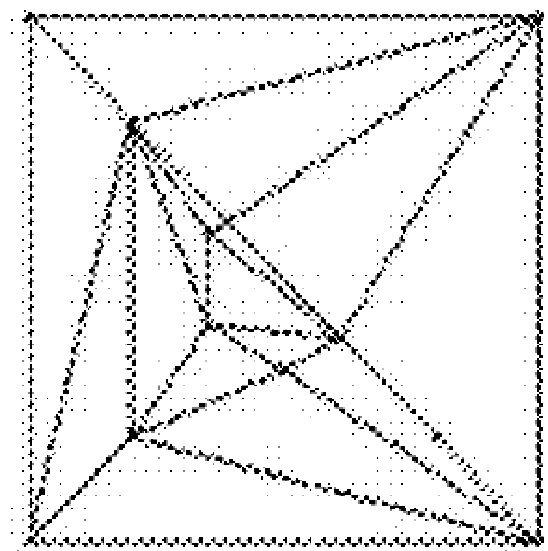


FIG. 12G

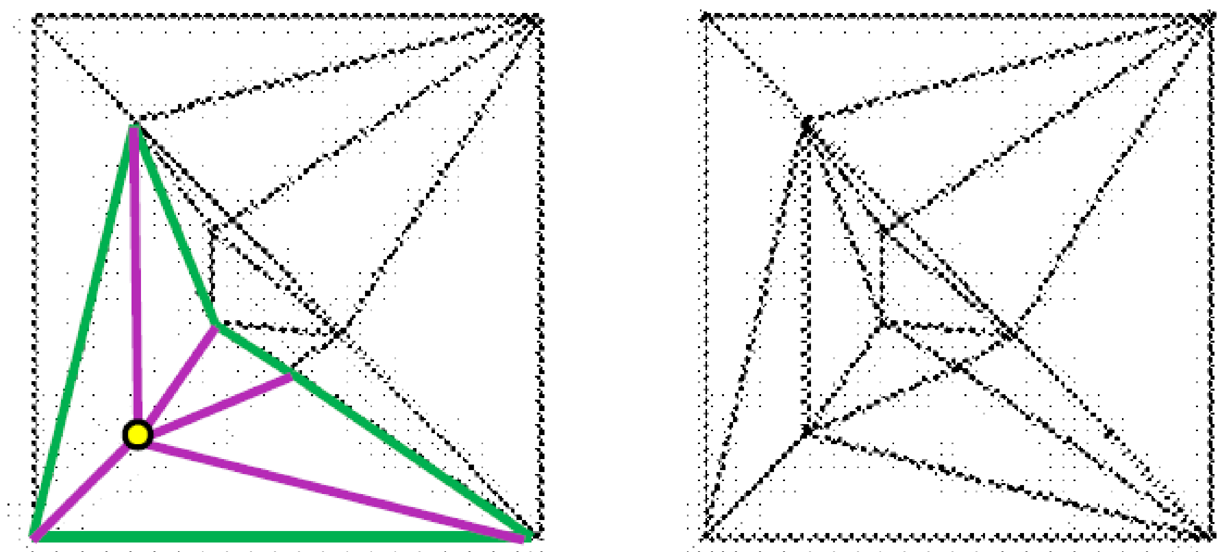


FIG. 12F Prior Art FIG. 12G

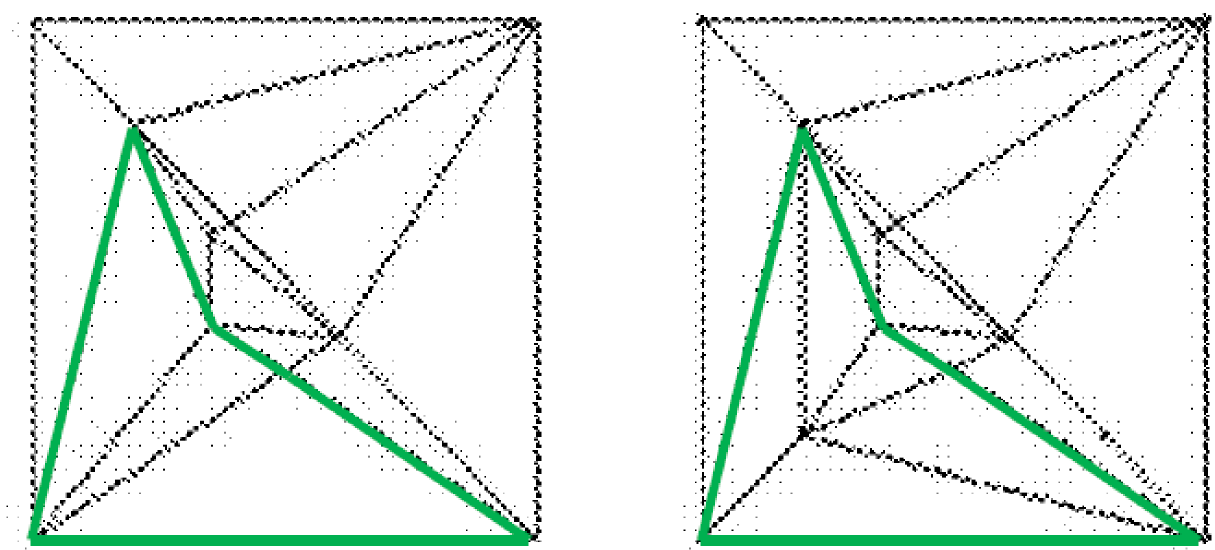


FIG. 12F Prior Art FIG. 12G

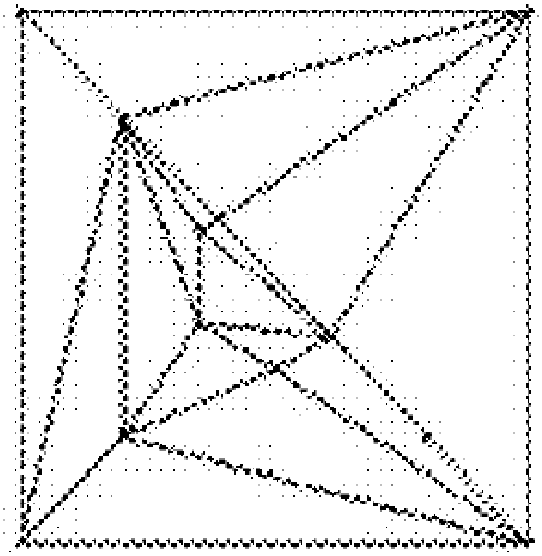


FIG. 12G

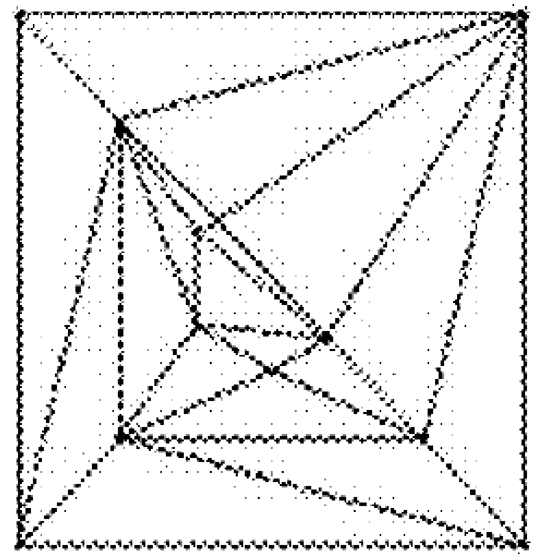


FIG. 12H

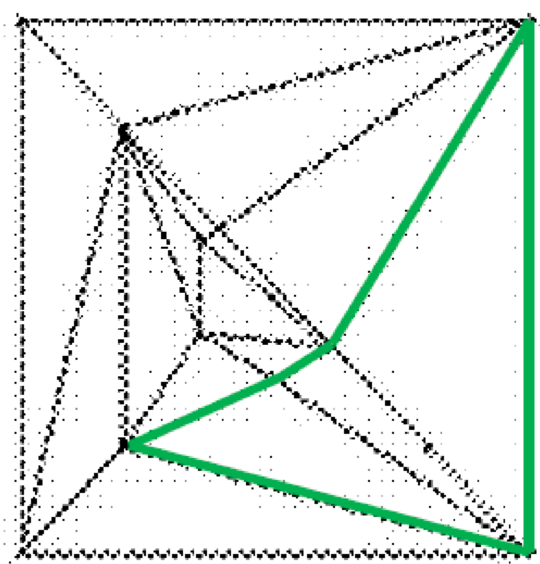


FIG. 12G

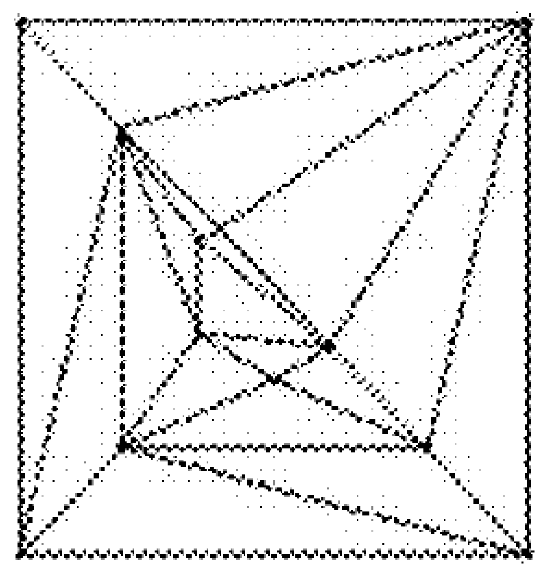


FIG. 12H

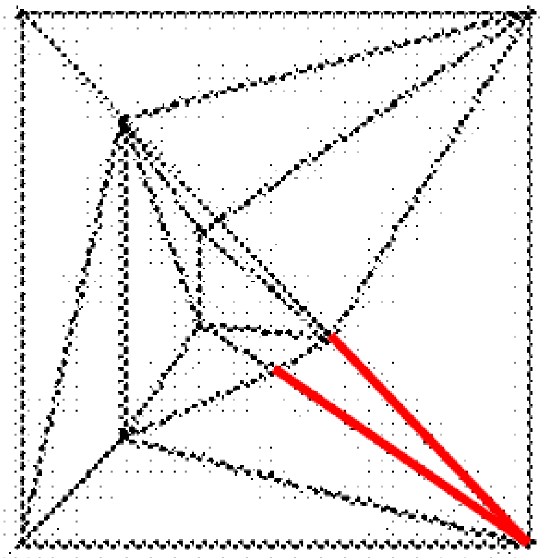


FIG. 12G

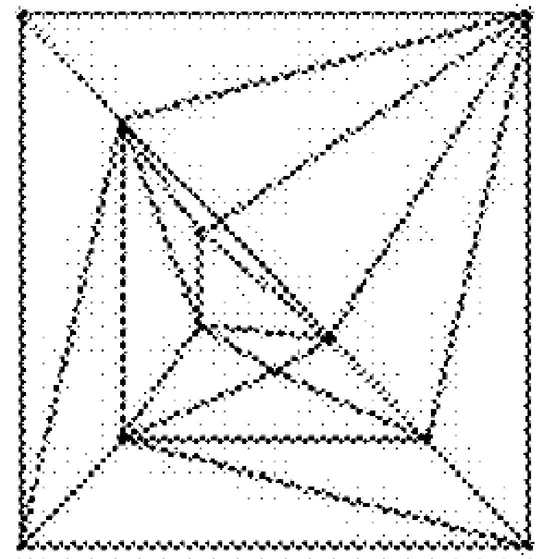


FIG. 12H

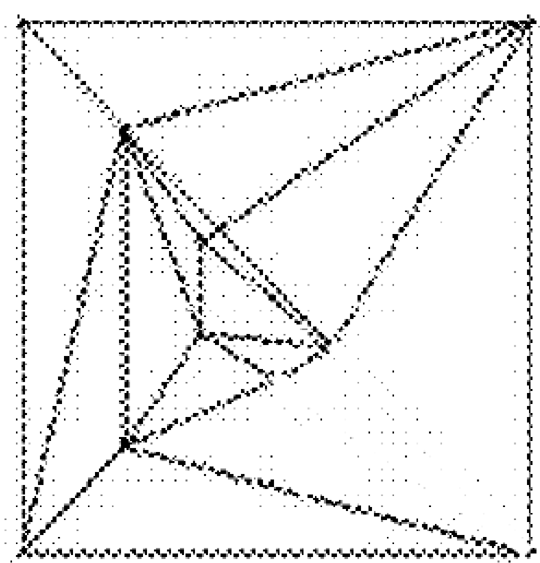


FIG. 12G

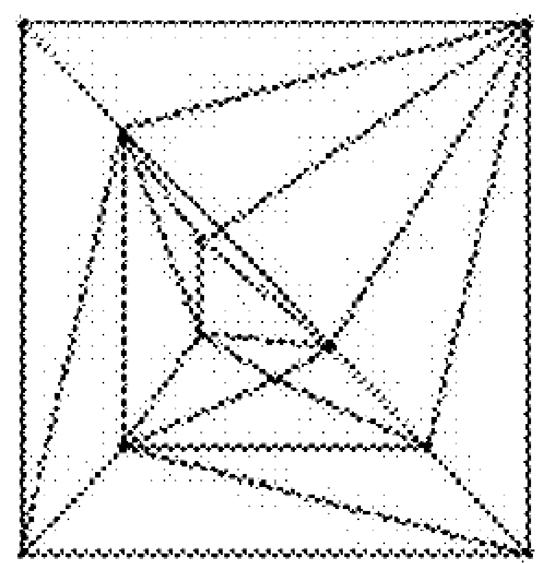


FIG. 12H

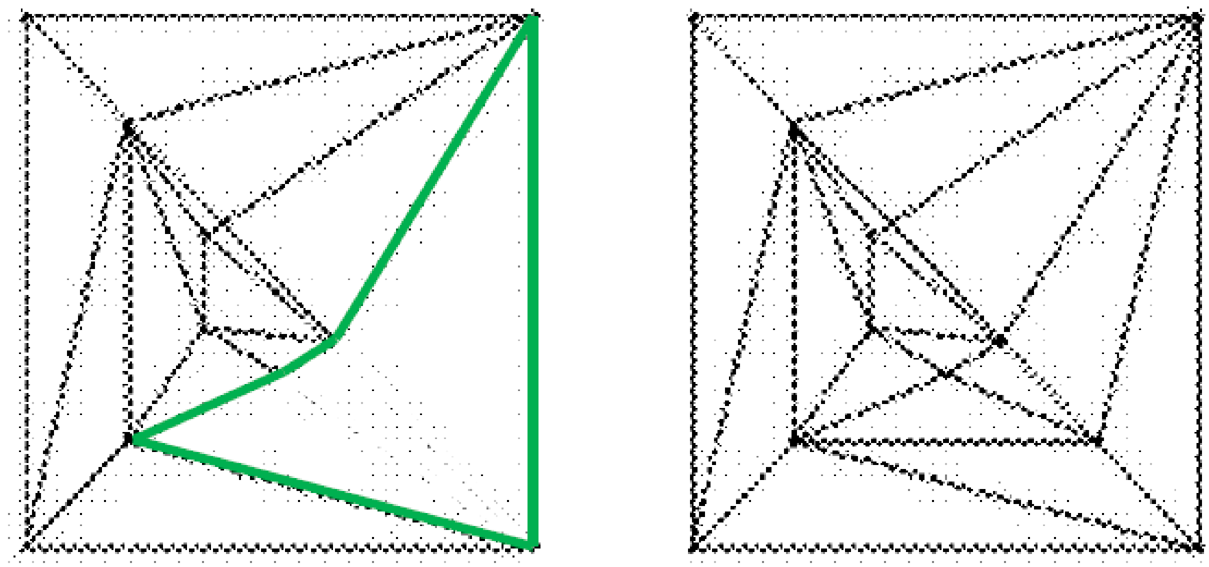


FIG. 12G FIG. 12H

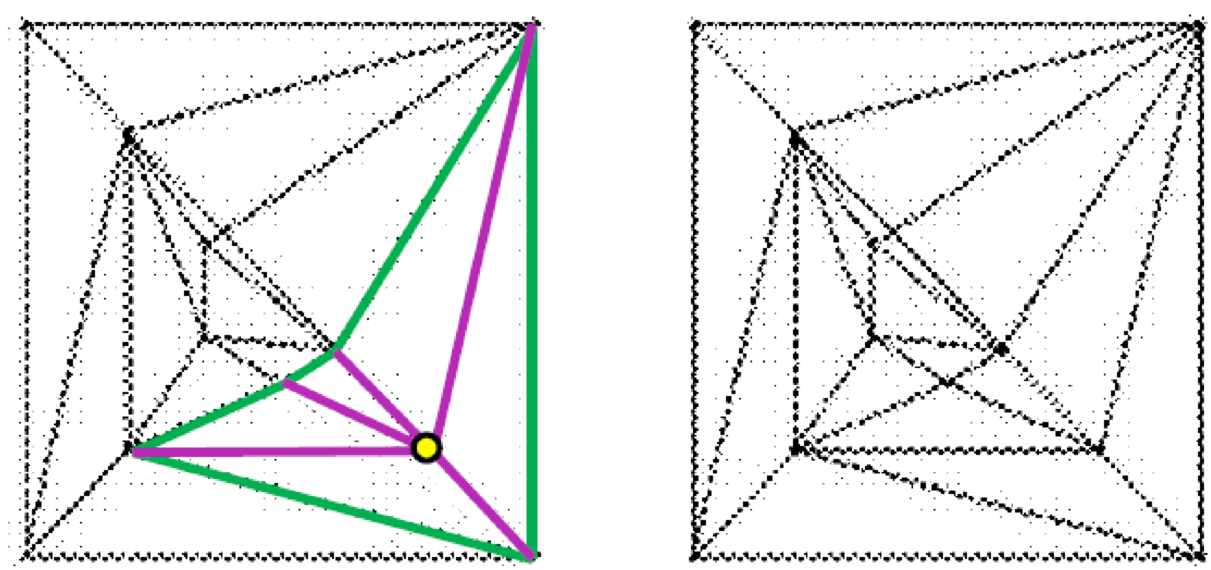


FIG. 12G FIG. 12H

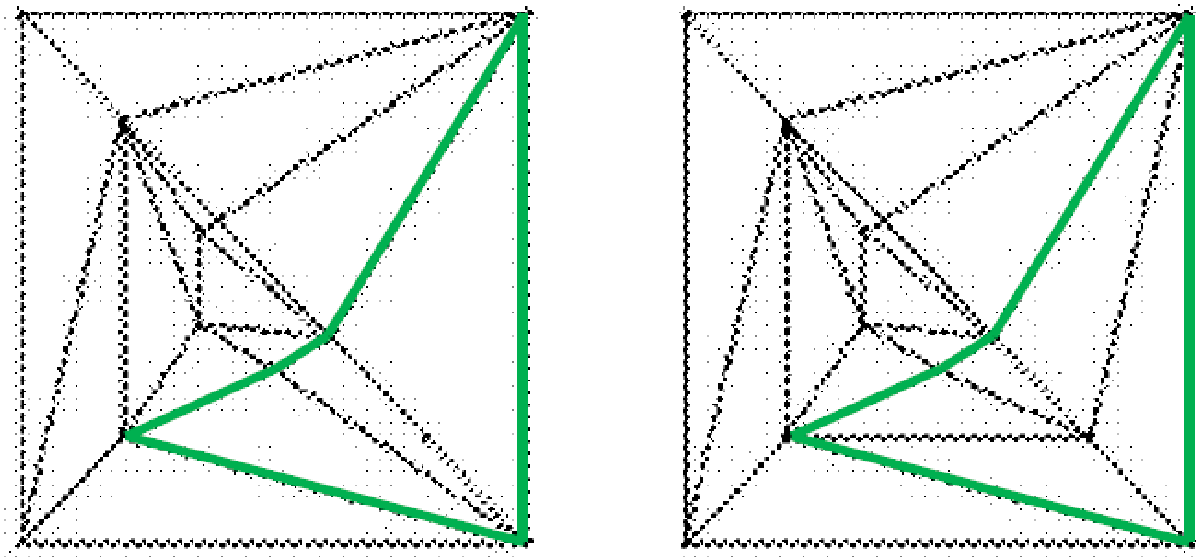


FIG. 12G FIG. 12H

This shows the conflict with the claim language.

Second, the progression from Figure 12F to Figure 12G does not split the triangle into an upper, middle, and lower portion, again as required in the claim language, as described above. The same problems occur in the progression from Figure 12F to Figure 12H.

iii. "searching neighboring triangles of the last triangle pair that holds the last intersection point."

Again, NSS argues that the phrase should be given its ordinary meaning, and Aliaga opines, with no opposition, that the phrase does not inform a POSITA of the scope of the invention. The disputed term appears in the limitation as follows:

> searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed.

('961 patent, 9:28-33; '105 patent, 8:54-63.) Aliaga points to several unanswered questions about this language: (1) whether the phrase requires searching repeatedly or iteratively or merely once; (2) what the "last triangle pair" or "last intersection point" is; (3) how one can "extend an

United States District Court
Northern District of California

intersection line" when in some cases it is not possible, as Aliaga demonstrates. (Dkt. 36-1 (Aliaga Dec. ¶ 26).)

In response, NSS argues that the language of the patents is clear and simply points to the patent language. NSS argues that the Figure 4 does not show an iterative process as Autodesk claims and points out that the threshold requirement for building the intersection line is to "search for the first not-traveled intersection point." (Dkt. 31-2 ('961 Patent, Figure 4).) In Figure 4, the method proceeds to the "extend intersection line" step only so long as "the first intersection line is found" and the method is search "for the first not-traveled intersection point." Figure 4 and the language are clear on this first issue regarding whether the process is iterative or not.

However, NSS has no response to the other specific questions that Aliaga poses other than to argue that the language is clear. Given Aliaga's uncontroverted opinion and specific questions raising ambiguity, NSS cannot prevail. NSS argues that the PTO amended the claim in such a way that clarified the term, and thus the Court should give deference to the PTO. 35 U.S.C. § 285(a); *see also Microsoft Corp.*, 564 U.S. at 95-96. However, as Autodesk points out, NSS raises this argument for the first time in its reply and did not cite this claim history in the joint claim construction pleading, both in violation of the Local Rules. *See* Civ. L.R. 7-3 (describing appropriate contents of reply); *California Sportfishing Prot. All. v. Pac. States Indus., Inc.*, 2015 WL 5569073, at *2 (N.D. Cal. Sept. 22, 2015) (finding that "reply brief improperly raises brand new arguments not stated in the opening brief" and refusing to consider those arguments for purposes of deciding the motion); Pat. L.R. 4-5 (describing appropriate contents of claim construction briefs and reply); *Ericsson Inc. v. Intellectual Ventures I LLC*, 901 F.3d 1374, 1380 (Fed. Cir. 2018) (discussing and approving general patent law practice of "reject[ing] arguments raised for the first time in a reply.").

NSS argues that Autodesk should not be able to weaponize the Local Rules. However, the Local Rules exist for a reason – to provide parties with adequate notice to prepare a response. Even though NSS did not comply with the Local Rules in this regard, the Court nonetheless considers NSS's arguments on the merits and finds that the claim history does not provide the clarification suggested. The prosecution history does not show how or why the amendment answered the unanswered questions. The prosecution history does not provide a clear reason for the amendment that clarifies the underlying ambiguity. The PTO initially rejected Claim 1 as indefinite: "The nexus between 'extending the intersection lines' and 'searching neighboring triangles' is also not clearly set forth. The examiner is not able to ascertain the scope of the claimed invention." (Dkt. 37-1 (Ex. A at page 4).) In response, NSS added the following language: "building intersection lines starting with and ending with . . . calculations for locating an intersection point, then searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed." (*Id.*) This added language does not answer the questions posed by Aliaga.

CONCLUSION

For the reasons set forth above, the Court finds that the claim language (1) "modified Watson method" and (2) "search neighboring triangles of the last triangle pair that holds the last intersection point" are indefinite.

IT IS SO ORDERED.

Dated: July 31, 2020

SALLIE KIM
United States Magistrate Judge

US010120961B2

(12) **United States Patent**
Cao

(10) **Patent No.: US 10,120,961 B2**
(45) **Date of Patent: *Nov. 6, 2018**

(54) **METHOD FOR IMMEDIATE BOOLEAN OPERATIONS USING GEOMETRIC FACETS**

(71) Applicant: **Shangwen Cao**, Montreal (CA)

(72) Inventor: **Shangwen Cao**, Montreal (CA)

(73) Assignee: **Nature Simulation Systems Inc.**, Montreal, Quebec (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **15/840,052**

(22) Filed: **Dec. 13, 2017**

(65) **Prior Publication Data**

US 2018/0113958 A1 Apr. 26, 2018

Related U.S. Application Data

(63) Continuation-in-part of application No. 15/207,927, filed on Jul. 12, 2016.

(51) **Int. Cl.**

G06F 17/50 (2006.01)
G06T 17/10 (2006.01)
G06F 17/10 (2006.01)
G06T 17/20 (2006.01)
G06F 9/30 (2018.01)

(52) **U.S. Cl.**

CPC ***G06F 17/50*** (2013.01); ***G06F 9/30029*** (2013.01); ***G06F 17/10*** (2013.01); ***G06T 17/10*** (2013.01); ***G06T 17/20*** (2013.01); *G06T 2210/21* (2013.01)

(58) **Field of Classification Search**

CPC G06F 17/50
USPC 345/420
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,825,369 A * 10/1998 Rossignac G06T 9/40 345/440
5,886,702 A * 3/1999 Migdal G06T 17/20 345/423
5,905,507 A * 5/1999 Rossignac G06T 9/40 345/440
5,929,860 A * 7/1999 Hoppe G06T 17/20 345/419

(Continued)

OTHER PUBLICATIONS

Landier, "Boolean operations on arbitrary polyhedral meshes", 24th International Meshing Roundtable (IMR24), 2015.

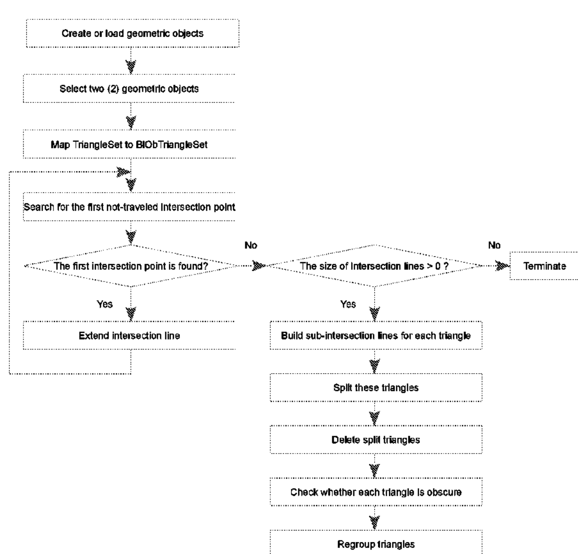
(Continued)

Primary Examiner — Jason C Olson

(57) **ABSTRACT**

A method for performing Boolean operations using a computer to create geometric models from primary geometric objects and their facets, comprises mapping rendering facets to extended triangles that contain neighbors; building intersection lines, splitting each triangle through which an intersection line passes, determining each facet is visible or obscure, and regrouping the facets to form one or more geometric objects. This method does not utilize the most popular data structures CSG and B-REP in CAD/CG/Solid Modeling systems, but has the advantages of both CSG and B-REP: easy to implement and flexible. Additionally it is a united method for solid modeling and surface modeling systems, and it is able to generate variant and editable models.

20 Claims, 6 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

5,945,996	A *	8/1999	Migdal	G06T 17/20	345/420
5,977,987	A *	11/1999	Duluk, Jr.	G06T 15/10	345/441
6,016,153	A *	1/2000	Gueziec	G06T 9/40	345/441
6,031,548	A *	2/2000	Gueziec	G06T 17/20	345/440
6,078,331	A *	6/2000	Pulli	G06T 17/20	345/423
6,172,679	B1 *	1/2001	Lim	G06T 15/40	345/421
6,191,787	B1 *	2/2001	Lu	G06T 17/05	345/418
6,307,555	B1 *	10/2001	Lee	G06T 17/20	345/421
6,573,895	B1 *	6/2003	Carter	G06T 15/40	345/423
6,587,105	B1 *	7/2003	Stam	G06T 17/20	345/423
6,850,638	B1 *	2/2005	Lounsbery	G06T 17/205	345/420
6,989,830	B2 *	1/2006	Stollnitz	G06T 17/20	345/420
7,127,380	B1 *	10/2006	Iverson	G06F 17/5018	703/2
7,133,043	B1 *	11/2006	Hollasch	G06T 15/06	345/421
7,209,137	B2 *	4/2007	Brokenshire	G06T 15/00	345/420
7,324,105	B1 *	1/2008	Moreton	G06T 17/20	345/420
7,366,581	B2 *	4/2008	Hill	G06T 17/10	700/118
7,408,553	B1 *	8/2008	Toksvig	G06T 11/40	345/423
7,589,720	B2 *	9/2009	Zhou	G06T 17/20	345/419
8,345,042	B2 *	1/2013	Kataoka	G06K 9/00214	345/420
8,547,395	B1 *	10/2013	Hutchins	G06T 11/40	345/611
9,401,046	B2 *	7/2016	Munkberg	G06T 17/205	
9,721,363	B2 *	8/2017	Williams	G06T 11/203	

OTHER PUBLICATIONS

Rappoport et al, "Interactive Boolean Operations for Conceptual Design of 3-D Solids", Institute of Computer Sciences, The Hebrew University, 1997.

E. Gursoz et al.,"Non-Regularized Boolean Set Operations on Non-Manifold B-Representation Objects", Computer-Aided Design 23 (1991) Jan./Feb. No. 1 London, GB.

D.F. Watson, "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes", The Computer Journal 24 (2) 1981.

* cited by examiner

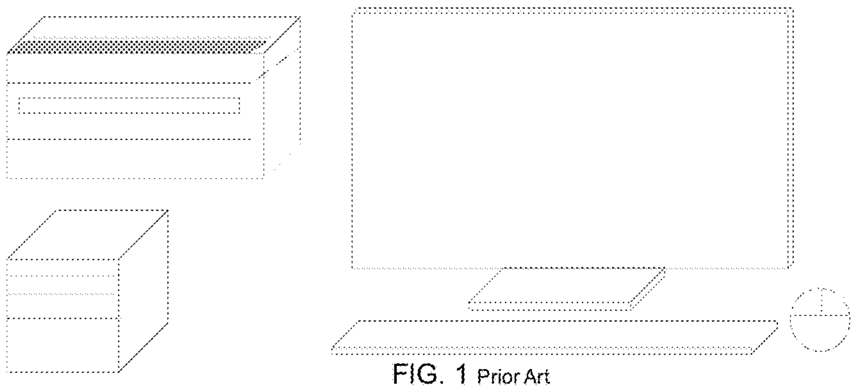


FIG. 1 Prior Art

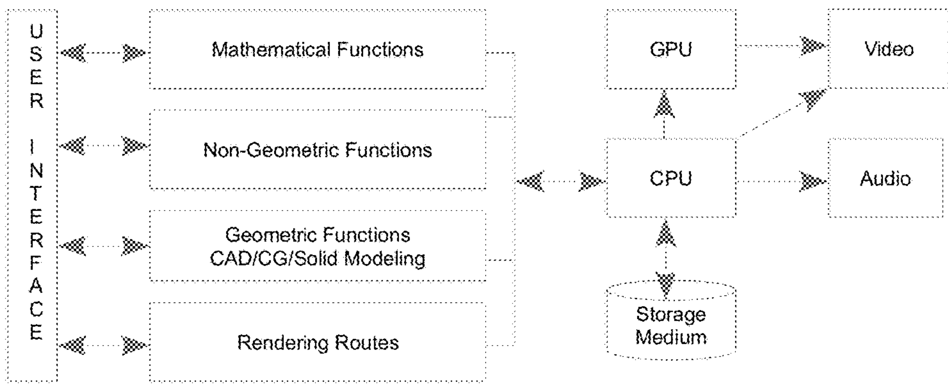


FIG. 2A Prior Art

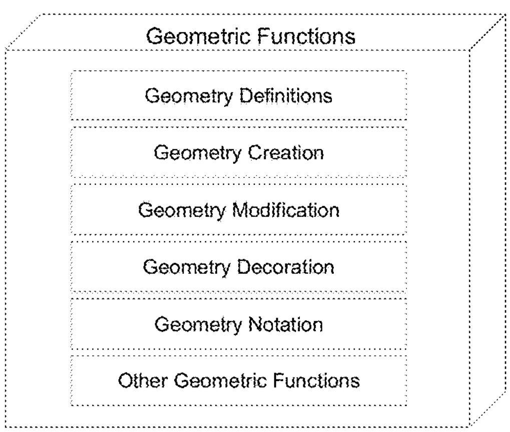


FIG. 2B Prior Art

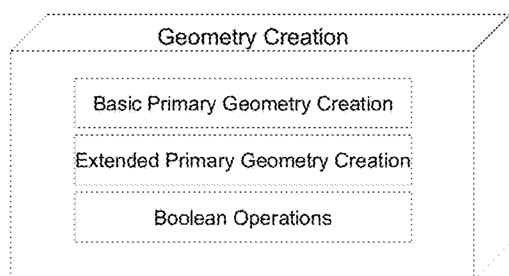


FIG. 2C Prior Art

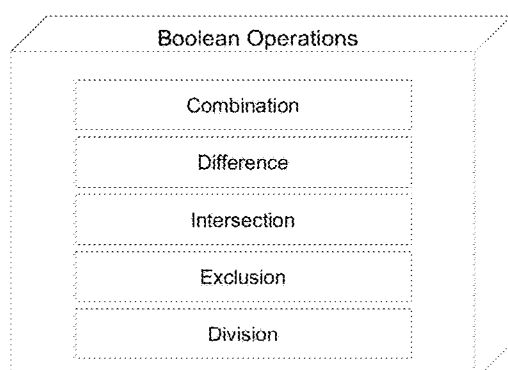


FIG. 2D Prior Art

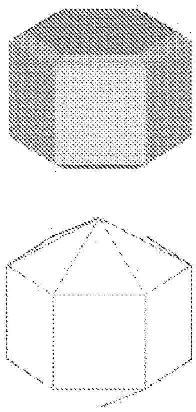


FIG. 3A Prior Art

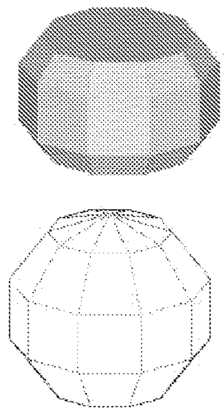


FIG. 3B Prior Art

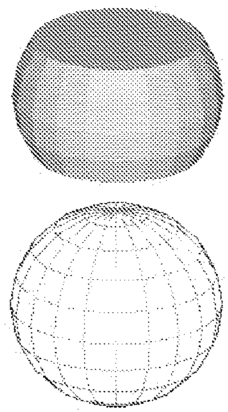


FIG. 3C Prior Art

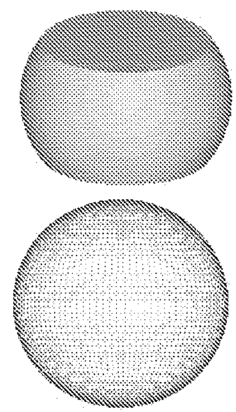


FIG. 3D Prior Art

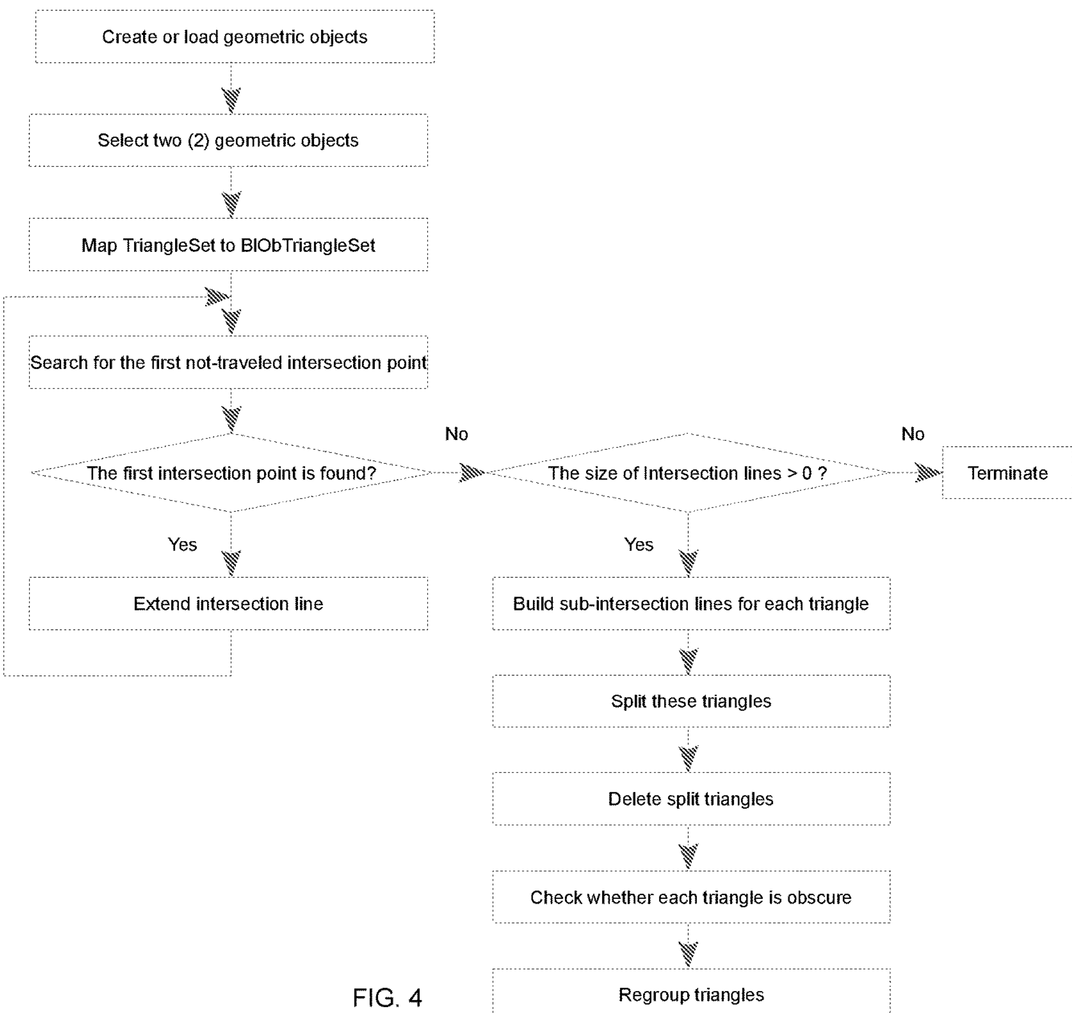


FIG. 4

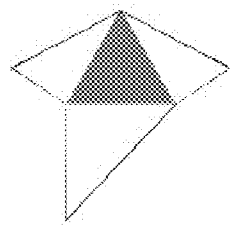


FIG. 5

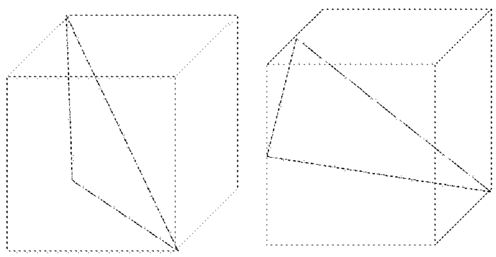


FIG. 6A

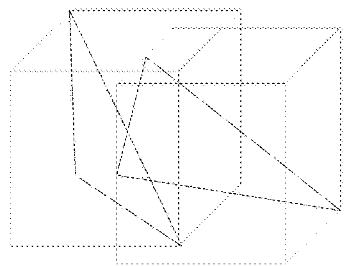


FIG. 6B

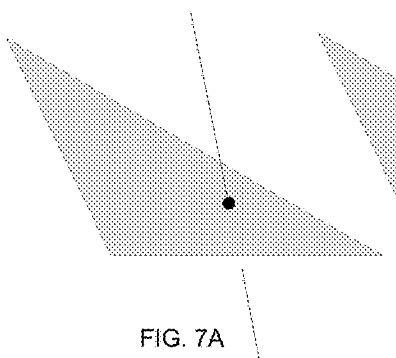


FIG. 7A

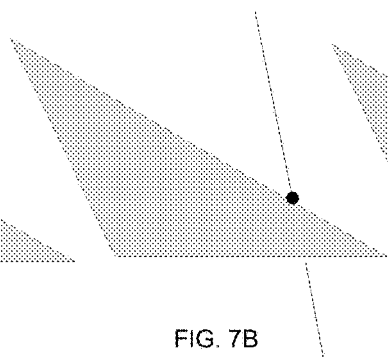


FIG. 7B

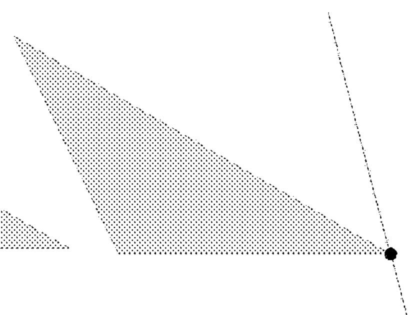


FIG. 7C

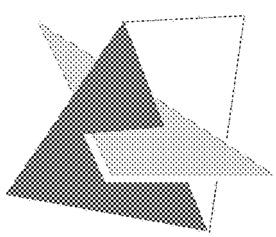


FIG. 8A

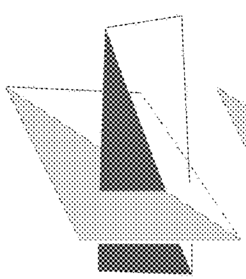


FIG. 8B

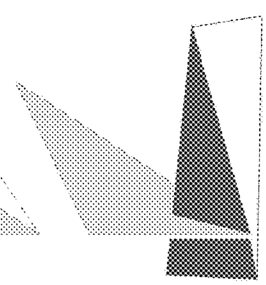


FIG. 8C

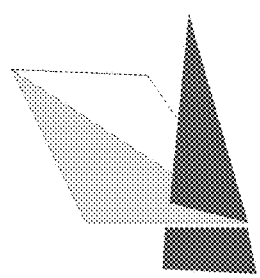


FIG. 8D

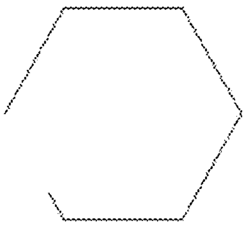


FIG. 9A

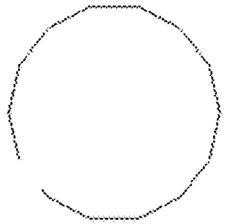


FIG. 9B

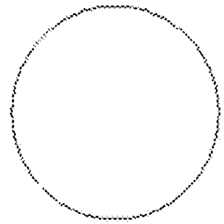


FIG. 9C

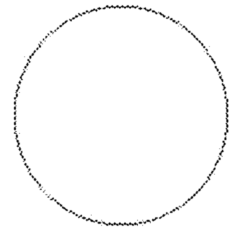


FIG. 9D

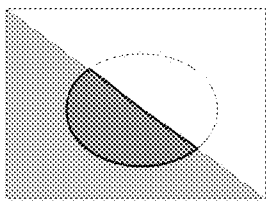


FIG. 10A

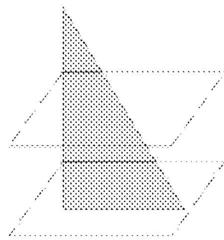


FIG. 10B

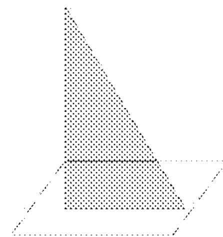


FIG. 10C

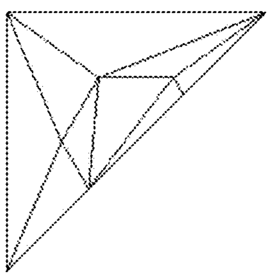


FIG. 11A

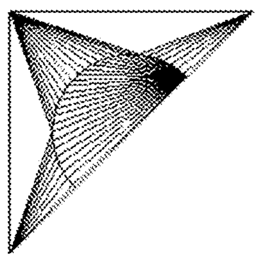


FIG. 11B

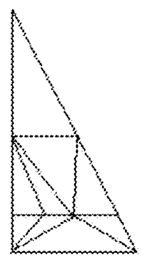


FIG. 11C

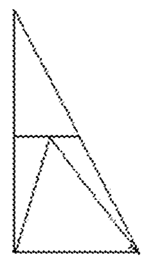


FIG. 11D

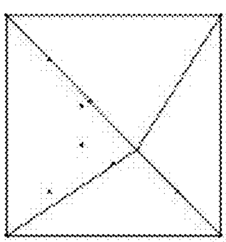


FIG. 12A Prior Art

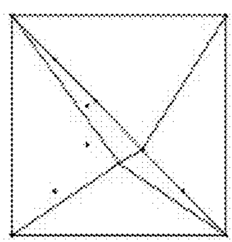


FIG. 12B Prior Art

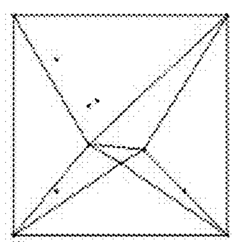


FIG. 12C Prior Art

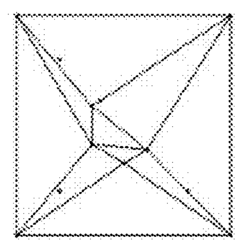


FIG. 12D Prior Art

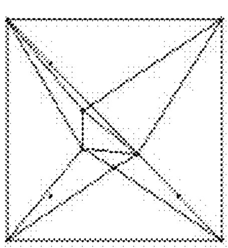


FIG. 12E Prior Art

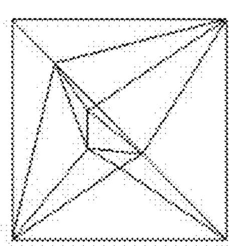


FIG. 12F Prior Art

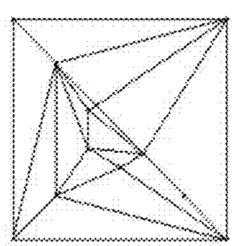


FIG. 12G

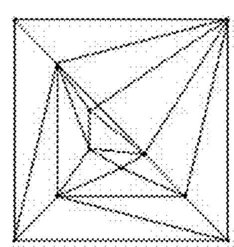


FIG. 12H

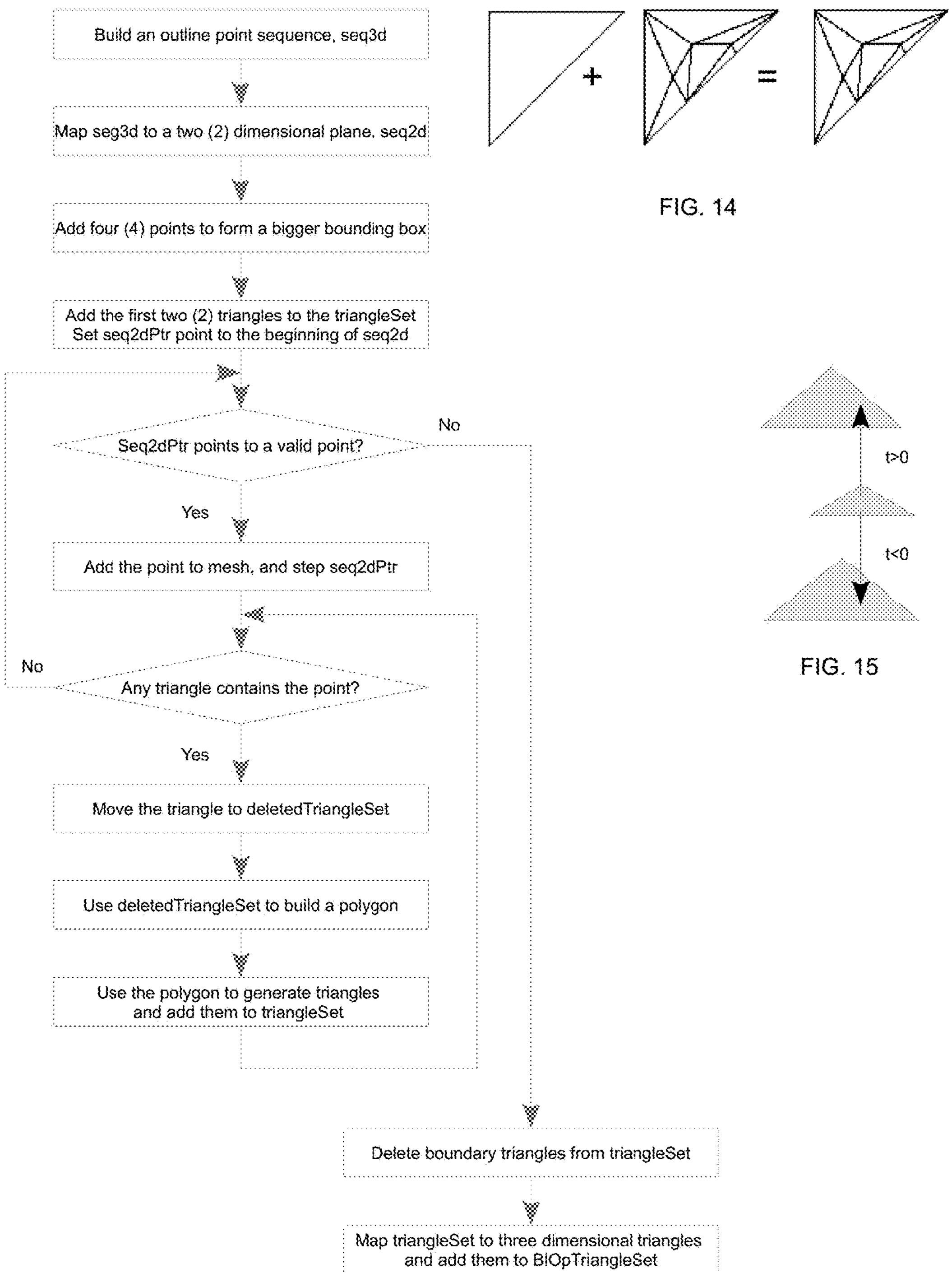


FIG. 14

FIG. 15

FIG. 13 Prior Art except the first two (2) steps , the last one, and the condition Any triangle contains the point.

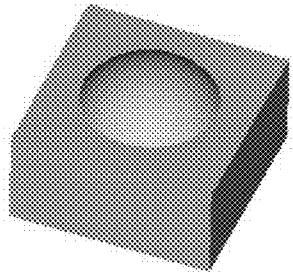


FIG. 16A

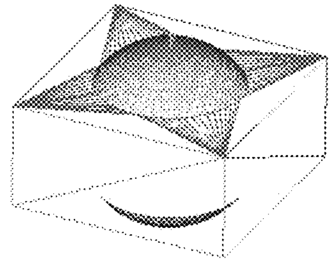


FIG. 16F

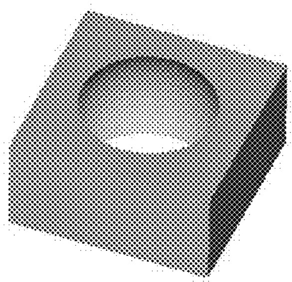


FIG. 16B

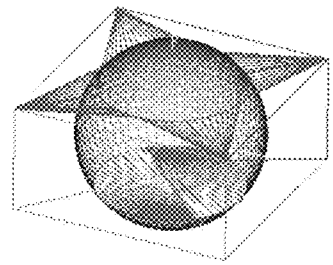


FIG. 16G

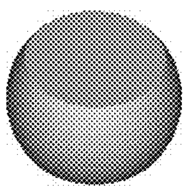


FIG. 16C

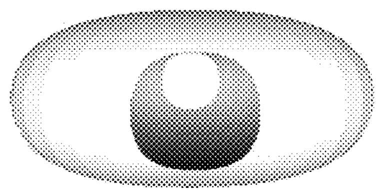


FIG. 17

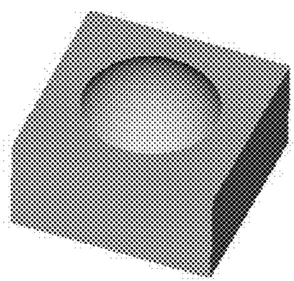


FIG. 16D

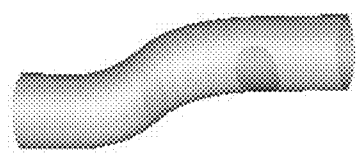


FIG. 18

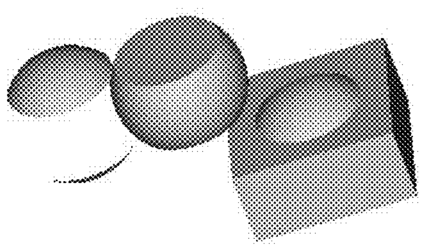


FIG. 16E

METHOD FOR IMMEDIATE BOOLEAN OPERATIONS USING GEOMETRIC FACETS

BACKGROUND

Field of the Invention

This invention provides an immediate Boolean operation method for building three (3) dimensional geometric models from primary geometric objects to Computer Aided Design, Computer Graphics, Solid Modeling systems, and Surface Modeling systems, which are widely used in product design, manufacturing, and simulation. Mechanic industry, culture and sports, everywhere there are geometric shapes, may have CAD/CG applications.

Related Art

Computer hardware is so highly developed that even an ordinary Personal Computer may be used to install and run a commercial CAD/CG system, which normally has Boolean operation functions including AND, OR, and NOT. PC components comprise input devices, such as a mouse and a keyboard, a main machine, a screen, and a printer. The software system contains geometric and non geometric functions. FIG. **1** shows the main PC components and FIGS. **2**A through **2**D depict a typical CAD/CG software system architecture.

Boolean operations provide a general process of building complex solid geometric objects from different geometric shapes, which include primary geometric objects, swept or extruded objects, to CAD/CG/Solid Modeling systems. Lee applied Boolean operations to divide surface [Lee U.S. Pat. No. 6,307,555].

Boolean operations may rely on Constructive Solid Geometry, CSG, to record primary geometric objects and operation sequence in a hierarchic way, which technically is easy to implement, whereas Boundary Representation, B-REP, is regarded as a more flexible way that supports more geometric object types like extended geometries [Gursoz, 1990].

This invention presents five (5) Boolean operation commands: combination, intersection, exclusion, difference, and division, which directly work on triangles decomposed from geometric facets used for rendering functions and do not require the data structure Constructive Solid Geometry or Boundary Representation. The data structures defined in this invention are a few of simple classes, the algorithms incorporated in this invention are concise and easy to implement, and the five (5) commands allow the user to create geometric models not only by selecting the types of geometric objects but also by defining their facets. FIG. **3** presents that a box with 6 facets and a sphere with different facets make distinct results.

Although the five (5) commands are designed for solid modeling and surface modeling, surface trimming command is incorporated in this invention provides an alternative for surface modeling and it identifies whether a facet is obscure in a different way.

This invention presents data structures and algorithms differ from CSG and B-REP, and the algorithms incorporated in this invention include triangle-triangle intersections, splitting triangles with sub-intersection lines, identifying whether a facet is obscure, and regrouping triangles to form geometric models.

DISCLOSURE OF THE INVENTION

This invention provides a set of data structures and algorithms for performing Boolean operations, which are

used to build complex geometric models and work directly on triangles decomposed from geometric facets used as rendering data by computer hardware and rendering functions like OpenGL libraries. A geometric shape, for example, a sphere, a cone, a cylinder, a box, triangular facets, an extruded or swept object, and a surface patch, is triangulated to build a set, noted as TriangleSet, for displaying. When two geometric shapes are selected for performing a Boolean operation, neighboring triangles will be added to each triangle in TriangleSet to form another set for each of the shapes, BIOpTriangleSet.

The second step of a Boolean operation this invention described is to search and build intersection lines between triangle sets. It starts with finding the first pair of intersecting triangles: this system builds an axis aligned minimum bounding box for each triangle and checks whether two bounding boxes overlap to decide if edge-triangle intersection needs to be calculated. Once the edge-triangle intersection point(s) falls inside a triangle, this system completes the searching task and stores the point data into an intersection line set.

To extend the current intersection line, this method traces neighboring triangles and calculates edge-triangle intersection points until the intersection line becomes closed.

The third step of a Boolean operation this invention described is to split triangles. Each segment of the intersection lines references two (2) triangles, each of the triangles has at least one sub-intersection line that contains one or more segments, which divide a triangle into three (3) or more smaller triangles. After splitting the triangles, the original triangles are removed, and those smaller triangles are added to the BIOpTriangleSet.

The fourth step of a Boolean operation this invention described is to decide each triangle is obscure or visible. If a triangle is enclosed by other triangles, it is obscure. A triangle is visible means it is outside another object.

The fifth step of a Boolean operation this invention described is to regroup the triangles: some of them have to be removed and some need to be put together, and there are five (5) cases for regrouping.

The final step of a Boolean operation this invention described is to map BIOpTriangleSet to TriangleSet.

The process of the said surface trimming command contains six (6) steps, too. Initially, this system maps a surface to a BIOpTriangleSet and one of its trimming contours to an extruded shape to form a BIOpTriangleSet. Step two (2), three (3), and six (6) are the same as that of the Boolean operations. Step four (4) checks a triangle is to the left or right side of the trimming contour to decide whether it is necessary to be reserved. The regrouping function, step five (5), deletes only left side or right side triangles when the system trims a surface.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows the main personal computer components, which generally contain a main machine, input devices including mouse and keyboard, a display, and a printer. A highly developed CAD/CG system can run on a PC machine.

FIGS. **2**A through **2**D describe a software architecture in which a CAD/CG/Geometric Modeling system uses Boolean operations and surface trimming to build geometric models.

FIG. **3** represents that distinct facets make various results even their original geometric object types and sizes are the same: the left side example has less facets and the right side has more facets. In these examples, Boolean intersection operations work on a box and a sphere.

FIG. **4** is a flowchart for immediate Boolean operations using geometric facets.

FIG. **5** depicts that a triangle has three (3) neighbors. Given a triangle and its two vertices, there is one and only one neighboring triangle in solid models.

FIGS. **6**A and **6**B show two minimum bounding boxes do not overlap and two boxes overlap each other. Each triangle virtually has a minimum bounding box. If two boxes do not overlap, the triangles contained in the two boxes do not intersect. If the boxes overlap, edge-triangle intersection calculation is required.

FIGS. **7**A through **7**C depict three (3) edge-triangle intersection cases: an intersection point falls inside a triangle, an intersection point locates on an edge of a triangle, an intersection point is a vertex of a triangle.

FIGS. **8**A through **8**D show the searching candidate set, which allows the system to traverse next triangle for extending intersection lines by conducting edge-triangle calculation. Triangles filled with colors are the last pair of triangles that intersect each other, the triangles not filled are referenced by the member m_NeigTri of the data structure Triangle3dEx, which guides the system searching a minimum set of triangles when building intersection lines. The set contains one triangle, two triangles, or zero.

FIGS. **9**A through **9**D show four (4) examples of intersection lines. A box intersects a sphere, which has different facet numbers.

FIGS. **10**A through **10**C give three (3) examples of sub-intersection lines in darker color. FIG. **10**A has one (1) sub-intersection line, **10**B two (2), and **10**C one (1).

FIGS. **11**A through **11**D show four (4) examples that sub-intersection lines divided a triangle into a set of triangles.

FIGS. **12**A through **12**H show a Delaunay mesh sequence in which each intersection point is inserted into the mesh step by step.

FIG. **13** is the flowchart of Delaunay mesh modified Watson method that created the sequence of FIGS. **12**A through **12**H.

FIG. **14** shows a triangle and its Delaunay mesh. The original triangle is removed and only the Delaunay mesh is reserved for late computations.

FIG. **15** shows t-Buffer where t may be negative and positive. If the size of negative t and positive t is balanced in t-Buffer, the triangle concerned is enclosed by another object and is obscure.

FIGS. **16**A through **16**E show five (5) examples of Boolean operations conducted with a box and a sphere. FIGS. **16**F and **16**G depict the internal mesh of two Boolean operation resultants: combination and exclusion.

FIG. **17** shows a contour line trims a closed surface, a deformed sphere, and generates two (2) holes.

FIG. **18** gives an example in which an extruded surface, a tube, is trimmed by a contour line and creates a hole.

DETAILED DESCRIPTION

This invention defines these data structures: Point3dEx, Triangle3dEx, and BIOpTriangle3dSet that inherit Point3d, Triangle3d, Triangle3dSet storing facets for rendering geometric objects. When performing a Boolean operation, the system maps rendering facets to BIOpTriangle3dSet and all following processes focus on the members and attributes of BIOpTriangle3dSet. FIG. **4** is the flowchart describing the main procedure of Boolean operations conducted by the present invention. After a Boolean operation is completed, the system maps the resultant stored in BIOpTriangle3dSet to rendering facets.

Geometric Facets for Rendering

CAD systems render facets to represent a geometric object, such as a sphere, a cone, a box, a cylinder, an extruded or swept object. A facet may compose three (3) or more points, and facets are usually decomposed into triangles for easy calculations. A box has six (6) facets decomposed into twelve (12) triangles. A sphere may have eighteen (18) facets, composing twenty four (24) triangles. A sphere may also be rendered using more than one thousand (1,000) facets and triangles. FIG. **3** shows a sphere rendered with different facets. This method uses Triangle3dSet to note triangle set data structure for rendering a geometric object, it contains two (2) attributes: a three (3) dimensional point set and a triangle set, where Triangle3d references Point3d.

```
class Triangle3dSet
{
    DataSet<Point3d>    m_PointSet;
    DataSet<Triangle3d> m_TriangleSet;
};
class Triangle3d
{
    Point3d        *m_Points[3];
};
class Point3d
{
    DataTypeI m_X, m_Y, m_Z;
};
```

Triangles for Boolean Operations

The Boolean Operation method described in this invention defined three (3) key classes: BIOpTriangleSet, Triangle3dEx, and Point3dEx.

```
class BIOpTriangleSet
{
    DataSet<Point3dEx>        m_PointSet;
    DataSet<Triangle3dEx>     m_TriangleSet;
};
class Point3dEx : Point3d
{
        DataTypeII      m_ID; // position and sequence index
        DataTypeIII     m_X, m_Y, m_Z; // DataType III may
                        be different from DataTypeI
};
class Triangle3dEx : Triangle3d
{
        Point3dEx       *m_Points[3];
        DataTypeII       m_ID;
        Plane            m_Plane;
        DataTypeIV       m_Normal[3];
        Triangle3dEx    *m_NeigTri[3]; // neighboring triangles
};
```

DataTypeII may be int, long, unsigned long, or other integer types. DataTypeIII is a floating point data type, such as float, double, even long double.

The class Triangle3dEx specifies each triangle may have three (3) neighboring triangles, and every triangle is stored just one (1) copy in BIOpTriangleSet. Given the box example, the simplest way it has twelve (12) triangles, even each of them has three (3) neighbors, BIOpTriangleSet still stores a total of twelve (12) triangles.

Technically Triangle3d may have the attribute m_Normal. If DataTypeI and DataTypeIV are the same type, for example, double, the attribute m_Normal can be inherited.

Data Mapping

The process of mapping Triangle3dSet to BIOpTriangleSet copies point set and triangle set from rendering statue and fills default attributes. Data mapping contains the following procedure:

1) Copy points from Triangle3dSet to BIOpTriangleSet and ensure there are not identical points.
2) Copy triangles from Triangle3dSet to BIOpTriangleSet.
3) For each triangle in BIOpTriangleSet, set its neighboring triangles.
4) Calculate the normal and build the plane equation for each triangle in BIOpTriangleSet.

Remark 1: Given two (2) points a and b, if $|x_a-x_b|<\varepsilon$ and $|y_a-y_b|<\varepsilon$ and $z_a-z_b|<\varepsilon$, where ε is a positive floating point number, for example 5.0e-16, then b is identical to a.

Remark 2: When mapping points from rendering data to BIOpTriangleSet, the system checks if there is an identical point in BIOpTriangleSet.

Remark 3: A triangle, which has three (3) points, defines a plane whose mathematical formula is ax+by+cz+d=0 and the class Plane internally records it as an array of four (4) numbers, such as double m_ABCD[4].

Remark 4: A triangle, if its three (3) points are not identical, always has a valid normal. Even it is related to m_ABCD, a separate copy makes things more clear and easy to handle later.

Remark 5: Every triangle has three (3) edges, when there are no duplicated points, it has three (3) neighboring triangles in solid models. FIG. **5** shows an example: a triangle filled with dark color and its three (3) neighbors. When concerning a surface patch for surface trimming, one or two neighbors of a triangle may be null.

The First Intersection Point

Every triangle has three (3) vertices, which define a minimum bounding box. This method adopted the concept of axis aligned minimum bounding box.

Given a pair of triangles, if their bounding boxes do not overlap, the two triangles have no intersection point; otherwise, this method carries out edge-triangle intersection calculation.

If an edge of a triangle T_a intersects with a plane defined by a triangle T_b and the intersection point pet falls inside T_b, then pet is the first intersection point. If pet is outside of T_b, then switch the triangle position in the pair, (T_a, T_b) changed to (T_b, T_a), and conduct edge-triangle intersection calculation.

Given the i-th edge of a triangle T_a, i∈[0, 2], its formula is: $p=p_i+t*(p_{(i+1)\%3}-p_i)$, and the plane defined by the triangle T_b, its formula is: ax+by+cz+d=0. If the two formulas have a solution, the edge intersects with the plane. If the edge-plane intersection point falls inside the triangle T_b, then the point is the edge-triangle intersection point.

Extending an Intersection Line

This method defines a data structure for recording an intersection point as PntEgTri:


```
class PntEgTri
{
    Triangle3dEx    *m_Tri0, *m_Tri1;
    DataTypeII      m_EdgeIndex;
    DataTypeII      m_PointPosi;
    Point3dEx       m_Point;
    Point3dEx       *m_PntGlobalIndexA, *m_PntGlobalIndexB;
};
```

According to the location of an intersection point on a triangle, a PntEgTri, simply said pet, can be classified into three (3) categories shown in FIGS. **7**A through **7**C.

1) The most popular case is the edge-triangle intersection, pet locates on an edge of triangle T_a and inside triangle T_b.
2) Edge-edge intersection, pet locates on an edge of triangle T_a and on an edge of triangle T_b.
3) Edge-vertex intersection, pet locates on an edge of triangle T_a and on a vertex of triangle T_b.

To extend an intersection line, this system catches next neighboring triangle(s) and checks edge-triangle intersection until the intersection line gets closed or all triangles are traversed.

Sub-Intersection Line

An intersection line passes through a set of triangles and divides each triangle into multi partitions. The segments of an intersection line inside a triangle make up a sub-intersection line. FIGS. **10**A through **10**C show three (3) examples in which the dark lines are sub-intersection lines. In practice, a triangle may have zero (0), one (1), two (2) or three (3) sub-intersection lines.

The following algorithm shows how to get a valid reference to a triangle that has at least one sub-intersection line:

```
for each intersection line
  for each intersection point, get the triangle references: (m_Tri0, m_Tri1)
    for each triangle of the triangle pair, if it is not split
      for each intersection line
        search and build a sub-intersection line
```

Given a valid triangle and an intersection line, to decide if a pet belongs to the sub-intersection line of the triangle, this method checks whether

1) pet is on an edge of the triangle,
2) or pet is inside the triangle,
3) or pet equals a vertex of the triangle.

Splitting a Triangle

Given a set of sub-intersection lines, to split a triangle, this method

1) Removes duplicated pets. If neighboring pets are identical, this method reserves just one copy.
2) Identifies the position of end pets: checks each pet locates on which edge of the triangle.
3) Splits the upper partition, down partition, and middle partition of the triangle where applicable.

Given a set of points on a plane that represents a partition of a triangle, to decompose the plane into a group of triangles, this invention modified Delaunay 2D mesh Watson method, which is published in 1981 [Watson, 1981].

A Delaunay 2D mesh has three (3) data sets: triangle set that holds the generated triangles, deleted triangle set that stores just deleted triangles, and polygon that records the outline of deleted triangle set.

The modified Delaunay 2D mesh method contains the following steps:

1) Build an outline point sequence that links sub-intersection lines and vertices of the triangle where applicable.
2) Map the three (3) dimensional point sequence to two (2) dimensional points according to the aspect of the plane.
3) Add four (4) points to form a bigger bounding box that encloses all the two (2) dimensional points.
4) Assume that one dialog line of the bounding box splits the box into two (2) triangles and add them into the triangle set.
5) Insert every point except bounding ones into the triangle set.
   a) For each point, check every triangle in the triangle set whether its circumcircle contains the point or the last segment passes through the triangle. If the condition is met, erase it from the triangle set and add it to the deleted triangle set.
   b) Use the deleted triangle set to extend the polygon and clear the deleted triangle set immediately.
   c) Use the polygon to generate triangles and add them to the triangle set.
6) Delete boundary triangles from the triangle set.
7) Map the triangle set to three dimensional triangles and add them to BIOpTriangleSet.

FIGS. **12**A through **12**H show a Delaunay 2D mesh sequence.

Deleting Split Triangles

In the above step, a split triangle got a mark. After all triangles have been traversed, this method deletes the marked triangles. FIG. **14** shows a deletion result.

Obscure Facets

Given two sets of triangles A and B, if A bounds a triangle of B, T_b, then T_b is obscure; if B bounds a triangle of A, T_a, then T_a is obscure.

To check whether a triangle T is bounded by an object O, this invention uses the following steps.

1) Calculate the centroid, c, of the triangle T
2) Build a line l: p=c+t*N, which starts from the centroid and passes along the normal N of the triangle T
3) For each triangle T_o of the object O, calculate line-plane intersection point. If there is a valid intersection point pet that falls inside the triangle T_o, then calculate t that is determined by centroid c and the pet, and add t to a depth buffer, butterT.
4) Check the size of negative t and positive t stored in butterT. If the two sizes are equal, then the triangle T is bounded and obscure.

When performing surface trimming, this system calls the followings procedure to determine whether a triangle is obscure.

1) Set the member m_ID of each Point3dEx of BIOpTriangleSet of the concerned surface patch to be 0.
2) Mark m_ID of Point3dEx of the intersection lines of the said patch in ascending or descending order, which is depending on whether the said line and trimming contour are in the same direction, for example, both of them are counterclockwise.

3) According to m_ID of the member m_Points of each triangle, decide whether it is a boundary triangle.
4) For each boundary triangle, decide it is to the left or right side of the trimming contour, and set its neighbors that are not boundary ones to be left or right.

Regrouping the Facets

This invention states five (5) kinds of Boolean operations, each of them has a different regrouping procedure.

The combination operation, logically it is OR, combines two solid geometric objects and generates a new object, which normally discards obscure partitions and reserves visible ones viewing from outside, has the following procedure.

1) Delete obscure triangles of object A.
2) Delete obscure triangles of object B.
3) Merge the triangles of object A and B.

The intersection operation, logically it is AND, which creates a solid geometric object using public sections of two geometric objects and discards any partitions of A and B outside the shared public sections, has the following procedure.

1) Delete NOT obscure triangles of object A.
2) Delete NOT obscure triangles of object B.
3) Merge the triangles of object A and B.

The exclusion operation, which builds a solid geometric object by removing public sections of two geometric objects and keeping not shared partitions, has the following procedure.

1) Copy object A's obscure triangles to a buffer, bufferA.
2) Delete obscure triangles from object A.
3) Copy object B's obscure triangles to object A.
4) Delete obscure triangles from object B.
5) Copy the triangles in bufferA to object B.
6) Reverse the normal of every obscure triangle of A and B.
7) Merge the triangles of the two objects.

The difference operation, which cuts geometric object A with another object B by removing any partitions of A inside B, has the following procedure.

1) Delete obscure triangles of object A.
2) Delete NOT obscure triangles of object B.
3) Reverse the normal of every triangle of object B.
4) Merge triangles of object A and B.

The division operation, which divides two solid geometric object A and B into three (3) objects, public sections of the two geometric objects, the NOT shared partitions of A and partitions of B, has the following procedure.

1) Copy object A's obscure triangles to a buffer, bufferA.
2) Copy object B's obscure triangles to bufferA.
3) Copy object A's obscure triangles to another buffer, bufferB.
4) Delete object A's obscure triangles.
5) Copy object B's obscure triangles to object A.
6) Delete object B's obscure triangles.
7) Copy object A's obscure triangles stored in bufferA to object B.
8) Reverse the normal of every obscure triangles of A and B.

Mapping to Rendering Facets

Once a Boolean operation is finished, this method maps BIOpTriangleSet to rendering triangles.

1) Each Point3dEx of BIOpTriangleSet is mapped to a Point3d of TriangleSet;

2) Each Triangle3dEx of BIOpTriangleSet is mapped to a Triangle3d of TriangleSet.

U.S. Patent Documents

U.S. Pat. No. 6,307,555 October 2001, Lee, 345/423.

Other Publications

"Non-Regularlized Boolean Set Operations on Non-Manifold B-Rep Objects", E. Gursoz et al., Carnegie Mellon University, Technical Report, 1990.

"Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes", D. F. Watson, The Computer Journal 24 (2) 1981.

What is claimed:

1. A method that performs immediate Boolean operations using geometric facets of geometric objects implemented in a computer system and operating with a computer, the method comprising:

mapping rendering facets to extended triangles that contain neighbors;

building intersection lines starting with and ending with searching for the first pair of triangles that hold a start point of an intersection line by detecting whether two minimum bounding boxes overlap and performing edge-triangle intersection calculations for locating an intersection point, then searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed;

splitting each triangle through which an intersection line passes using modified Watson method, wherein the modified Watson method includes removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion, and a middle portion;

checking each triangle whether it is obscure or visible for Boolean operations or for surface trimming;

regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets; and

mapping extended triangles to rendering facets.

2. The method of claim 1 wherein any Boolean operations, that use rendering facets of the geometric objects to create new geometric objects, including combination, intersection, exclusion, difference, and division, map rendering facets to extended triangles, build intersection lines, split each triangle through which an intersection line passes, check each triangle whether it is obscure or visible, regroup facets to form new extended triangle sets, and map extended triangles to rendering facets at the end of a Boolean operation or surface trimming without the data structure Constructive Solid Geometry and Boundary Representation.

3. The method of claim 1 wherein any Boolean operations that use rendering facets of the geometric objects to create new geometric objects, including combination, intersection, exclusion, difference, and division, map rendering facets to extended triangles, build intersection lines, split each triangle through which an intersection line passes, check each triangle whether it is obscure or visible, regroup facets to form new extended triangle sets, and map extended triangles to rendering facets at the end of a Boolean operation or surface trimming with the data structure Constructive Solid Geometry or Boundary Representation.

4. The method of claim 1 wherein building intersection lines uses the minimum bounding boxes to detect whether two facets do not overlap and carries out edge-triangle intersection calculations comprising the steps ensuring that the intersection points are exact and the intersection lines are not approximate curves: building the formula $p=p_i+t*(p_{(i+1)\%3}-p_i)$ for expressing the i-th edge of the triangle T_a that is one triangle inside the triangle pair, building the formula ax+by+cz+d=0 for recording the plane defined by the triangle T_b that is another triangle inside the triangle pair, and getting the solution of the two linear formulas.

5. The method of claim 1 wherein searching for the first pair of triangles and searching neighboring triangles calculate edge-triangle intersection and employee neighboring triangles ensuring that direct calculation of edge-edge intersection is replaced by verifying whether a point is on an edge of a triangle.

6. The method of claim 1 wherein splitting each triangle and builds Delaunay 2D mesh with modified Watson method that defines a triangle set, a deleted triangle set, and a polygon, dividing the triangle into different partitions even when the sub-intersection lines are not convex, comprising of: building an outline point sequence that links sub-intersection lines and vertices of the triangle where applicable; mapping the three (3) dimensional point sequence to two (2) dimensional points according to the aspect of the plane defined by the triangle; adding four (4) points to form a bigger bounding box that encloses all the two (2) dimensional points; assuming that one dialog line of the bounding box splits the box into two (2) triangles and adding them into the triangle set; inserting every point except bounding ones into the triangle set with the steps: for each point, checking every triangle in the triangle set whether its circumcircle contains the point or the last segment passes through the triangle, and when the condition is met, erasing it from the triangle set and adding it to the deleted triangle set, using the deleted triangle set to extend the polygon and clearing the deleted triangle set immediately, and using the polygon to generate triangles and adding them to the triangle set; and deleting boundary triangles from the triangle set; mapping the triangle set to three dimensional triangles and adding them to BIOpTriangleSet.

7. The method of claim 1 wherein checking each triangle whether it is obscure or visible further composing: calculating the centroid c of triangle T_a that belongs to geometric object A; building a line l: p=c+t*N passing through the centroid c and along the normal of T_a; for each triangle T_b of object B, checking whether l intersects with T_b at an interior point and adding t to a depth buffer, t-Buffer; and setting T_a to be obscure when the size of negative t equals to that of positive t in t-Buffer.

8. The method of claim 1 wherein the checking each triangle whether it is obscure or visible when trimming a surface patch with a trimming contour further composing: setting m_ID of regular points of BIOpTriangleSet of the concerned patch to be 0 and m_ID of points of the intersection lines of the said patch in ascending or descending order, which is depending on whether the said line and the trimming contour are in the same direction; according to m_ID of the member m_Points of each triangle, deciding whether it is a boundary triangle; for each boundary triangle, determining it is to the left or right side of the trimming contour, and setting its neighbors that are not boundary ones to be left or right.

9. The method of claim 1 wherein a Boolean operation that is a combination, an intersection, an exclusion, a difference, or a division, regroups facets for constructing its operational result using one or more steps of: deleting obscure or visible triangles of an object, copying obvious triangles of an object to a buffer or copying triangles from a buffer to an object, reversing the normal of each triangle of an object, and merging the triangles of the objects to form new extended triangle sets.

10. The method of claim 1 wherein the extended triangles are directly mapped to rendering facets for being displayed and providing data to next Boolean operations.

11. A computer system consisting of hardware and software that performs immediate Boolean operations using rendering facets of geometric objects, the system comprising:

a computer with input devices for entering data and commands, and a display device showing user interface, geometric objects, and additional data, having a medium storing geometric data and instructions that make up of a software system, or having a microchip or integrated circuit embedding partially or totally the instructions, and a processor that executes the steps of:

creating, modifying or loading primary geometric objects including swept and extruded ones and relocating them at different positions or orientations with input devices of the computer;

selecting two of the geometric objects;

mapping rendering facets to extended triangles that contain neighbors;

building intersection lines starting with and ending with searching for the first pair of triangles that hold a start point of an intersection line by detecting whether two minimum bounding boxes overlap and by performing edge-triangle intersection calculations for locating an intersection point, then searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed;

splitting each triangle through which an intersection line passes using modified Watson method, wherein the modified Watson method includes removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion, and a middle portion;

checking each triangle whether it is obscure or visible for Boolean operations or for surface trimming;

regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets; and

mapping extended triangles to rendering facets.

12. The system of claim 11 wherein any Boolean operations that use rendering facets of the geometric objects to create new geometric objects, including combination, intersection, exclusion, difference, and division, map rendering facets to extended triangles, build intersection lines, split each triangle through which an intersection line passes, check each triangle whether it is obscure or visible, regroup facets to form new extended triangle sets, and map extended triangles to rendering facets at the end of a Boolean operation or surface trimming with the data structure Constructive Solid Geometry and Boundary Representation.

13. The system of claim 11 wherein any Boolean operations that use rendering facets of the geometric objects to create new geometric objects, including combination, intersection, exclusion, difference, and division, map rendering facets to extended triangles, build intersection lines, split each triangle through which an intersection line passes, check each triangle whether it is obscure or visible, regroup facets to form new extended triangle sets, and map extended triangles to rendering facets at the end of a Boolean operation or surface trimming with the data structure Constructive Solid Geometry or Boundary Representation.

14. The system of claim 11 wherein searching for the first pair of triangles that hold a start point of an intersection line and searching neighboring triangles of the last triangle pair that hold the last intersection point composed the procedure for building an intersection line that usually repeats more than one time when building intersection lines use the minimum bounding boxes to detect whether two facets do not overlap and carries out edge-triangle intersection calculations comprising the steps ensuring that the intersection points are exact and the intersection lines are not approximate curves: building the formulae $p=p_i+t*(p_{(i+1)\%3}-p_i)$ for expressing the i-th edge of the triangle T_a that is one triangle inside the triangle pair, building the formula ax+by+cz+d=0 for recording the plane defined by the triangle T_b that is another triangle inside the triangle pair, and getting the solution of the two linear formulas.

15. The system of claim 11 wherein searching for the first pair of triangles and searching neighboring triangles calculate edge-triangle intersection and employee neighboring triangles ensuring that direct calculation of edge-edge intersection is replaced by verifying whether a point is on an edge of a triangle.

16. The system of claim 11 wherein splitting each triangle builds Delaunay 2D mesh with modified Watson method that defines a triangle set, a deleted triangle set, and a polygon, dividing the triangle into different partitions even when the sub-intersection lines are not convex, comprising of: building an outline point sequence that links sub-intersection lines and vertices of the triangle where applicable; mapping the three (3) dimensional point sequence to two (2) dimensional points according to the aspect of the plane defined by the triangle; adding four (4) points to form a bigger bounding box that encloses all the two (2) dimensional points; assuming that one dialog line of the bounding box splits the box into two (2) triangles and adding them into the triangle set; inserting every point except bounding ones into the triangle set with the steps: for each point, checking every triangle in the triangle set whether its circumcircle contains the point or the last segment passes through the triangle, and when the condition is met, erasing it from the triangle set and adding it to the deleted triangle set, using the deleted triangle set to extend the polygon and clearing the deleted triangle set immediately, and using the polygon to generate triangles and add them to the triangle set; and deleting boundary triangles from the triangle set; mapping the triangle set to three dimensional triangles and adding them to BIOpTriangleSet.

17. The system of claim 11 wherein checking each triangle whether it is obscure or visible further composing: calculating the centroid c of triangle T_a that belongs to geometric object A; building a line l: p=c+t*N passing through the centroid c and along the normal of T_a; for each triangle T_b of object B, checking whether l intersects with T_b at an interior point and adding t to a depth buffer, t-Buffer; and setting T_a to be obscure when the size of negative t equals to that of positive t in t-Buffer.

18. The system of claim **11** wherein checking each triangle whether it is obscure or visible when trimming a surface patch with a trimming contour further composing: setting m_ID of regular points of BIOpTriangleSet of the concerned patch to be 0 and m_ID of points of the intersection lines of the said patch in ascending or descending order, which is depending on whether the said line and the trimming contour are in the same direction; according to m_ID of the member m_Points of each triangle, deciding whether it is a boundary triangle; for each boundary triangle, determining it is to the left or right side of the trimming contour, and setting its neighbors that are not boundary ones to be left or right.

19. The system of claim **11** wherein a Boolean operation that is a combination, an intersection, an exclusion, a difference, or a division, regroups facets for constructing its operational result using one or more steps of: deleting obscure or visible triangles of an object, copying obscure triangles of an object to a buffer or copying triangles from a buffer to an object, reversing the normal of each triangle of an object, and merging the triangles of the objects to form new extended triangle sets.

20. The system of claim **11** wherein the extended triangles are directly mapped to rendering facets for being displayed and providing data to next Boolean operations.

* * * * *

US010109105B2

(12) **United States Patent**
Cao

(10) **Patent No.: US 10,109,105 B2**
(45) **Date of Patent: Oct. 23, 2018**

(54) **METHOD FOR IMMEDIATE BOOLEAN OPERATIONS USING GEOMETRIC FACETS**

(71) Applicant: **Shangwen Cao**, Montreal (CA)

(72) Inventor: **Shangwen Cao**, Montreal (CA)

(73) Assignee: **Nature Simulation Systems Inc.**, Montreal, Quebec (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 50 days.

(21) Appl. No.: **15/207,927**

(22) Filed: **Jul. 12, 2016**

(65) **Prior Publication Data**

US 2018/0018818 A1 Jan. 18, 2018

(51) **Int. Cl.**
G06T 17/10 (2006.01)
G06T 15/04 (2011.01)

(52) **U.S. Cl.**
CPC ***G06T 17/10*** (2013.01); ***G06T 15/04*** (2013.01); *G06T 2210/12* (2013.01)

(58) **Field of Classification Search**
CPC G06F 17/50
USPC 345/420
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,825,369 A * 10/1998 Rossignac G06T 9/40 345/440
5,886,702 A * 3/1999 Migdal G06T 17/20 345/423
5,905,507 A * 5/1999 Rossignac G06T 9/40 345/440
5,929,860 A * 7/1999 Hoppe G06T 17/20 345/419
5,945,996 A * 8/1999 Migdal G06T 17/20 345/420
5,977,987 A * 11/1999 Duluk, Jr. G06T 15/10 345/441
6,016,153 A * 1/2000 Gueziec G06T 9/40 345/441
6,031,548 A * 2/2000 Gueziec G06T 17/20 345/440
6,078,331 A * 6/2000 Pulli G06T 17/20 345/423
6,172,679 B1 * 1/2001 Lim G06T 15/40 345/421
6,191,787 B1 * 2/2001 Lu G06T 17/05 345/418
6,307,555 B1 * 10/2001 Lee G06T 17/20 345/421
6,573,895 B1 * 6/2003 Carter G06T 15/40 345/423

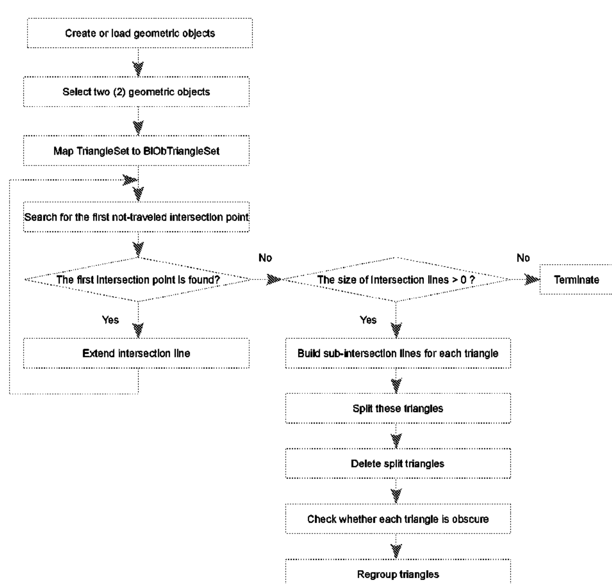
(Continued)

Primary Examiner — Jason C Olson

(57) **ABSTRACT**

A method for performing Boolean operations using a computer to create geometric models from primary geometric objects and their facets, comprises mapping rendering facets to extended triangles that contain neighbors, building intersection lines, splitting each triangle through which an intersection line passes, determining each triangle is obscure or visible, and regrouping the triangles to form one or more geometric objects. This method does not utilize the most popular data structures CSG and B-REP in CAD/CG/Solid Modeling systems, but has the advantages of both CSG and B-REP: easy to implement and so flexible that it can handle concave and convex geometric shapes, swept and extruded geometric objects, and it is able to generate variant and editable models.

20 Claims, 6 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

6,587,105	B1 *	7/2003	Stam	G06T 17/20 345/423
6,850,638	B1 *	2/2005	Lounsbery	G06T 17/205 345/420
6,989,830	B2 *	1/2006	Stollnitz	G06T 17/20 345/420
7,127,380	B1 *	10/2006	Iverson	G06F 17/5018 703/2
7,133,043	B1 *	11/2006	Hollasch	G06T 15/06 345/421
7,209,137	B2 *	4/2007	Brokenshire	G06T 15/00 345/420
7,324,105	B1 *	1/2008	Moreton	G06T 17/20 345/420
7,366,581	B2 *	4/2008	Hill	G06T 17/10 700/118
7,408,553	B1 *	8/2008	Toksvig	G06T 11/40 345/423
7,589,720	B2 *	9/2009	Zhou	G06T 17/20 345/419
8,345,042	B2 *	1/2013	Kataoka	G06K 9/00214 345/420
8,547,395	B1 *	10/2013	Hutchins	G06T 11/40 345/611
9,401,046	B2 *	7/2016	Munkberg	G06T 17/205
9,721,363	B2 *	8/2017	Williams	G06T 11/203

* cited by examiner

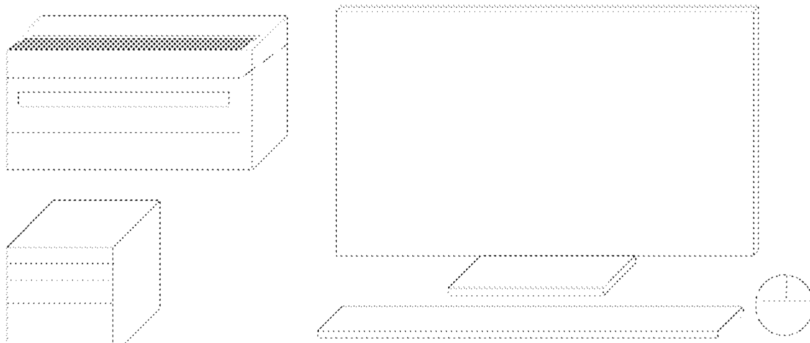


FIG. 1

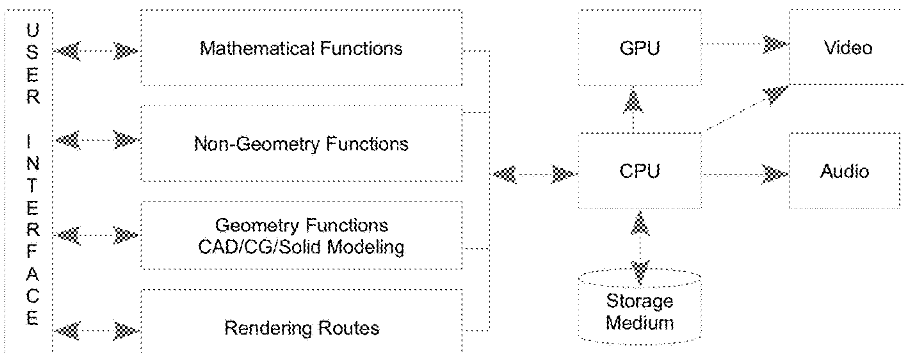


FIG. 2A

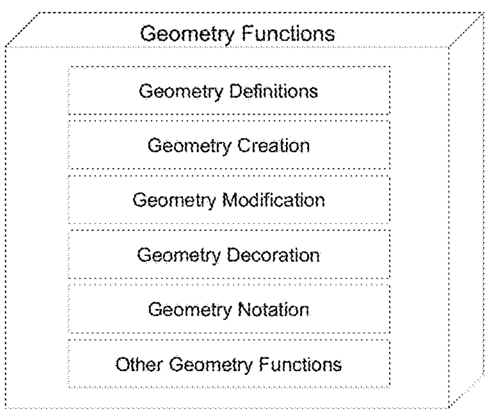


FIG. 2B

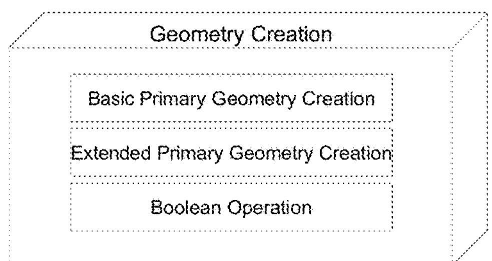


FIG. 2C

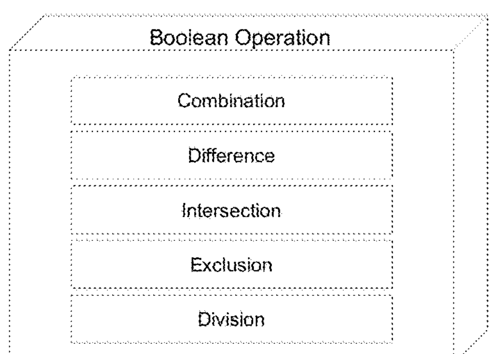


FIG. 2D

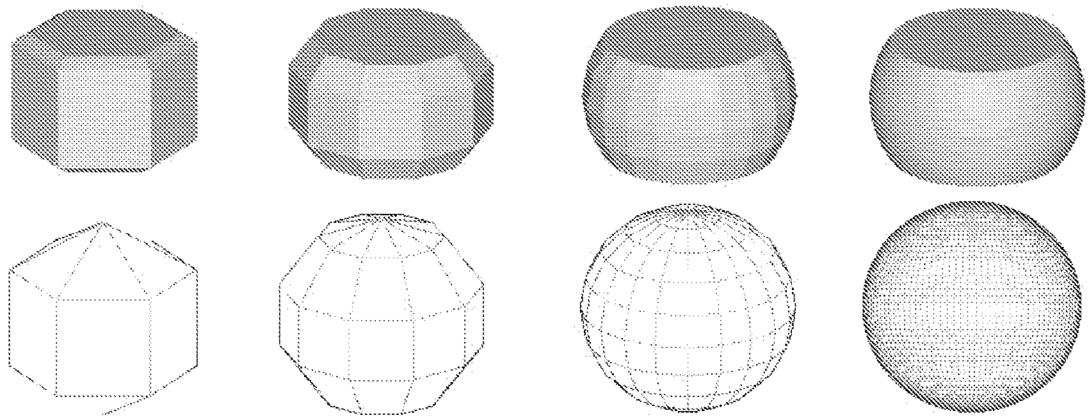


FIG. 3A Prior Art FIG. 3B Prior Art FIG. 3C Prior Art FIG. 3D Prior Art

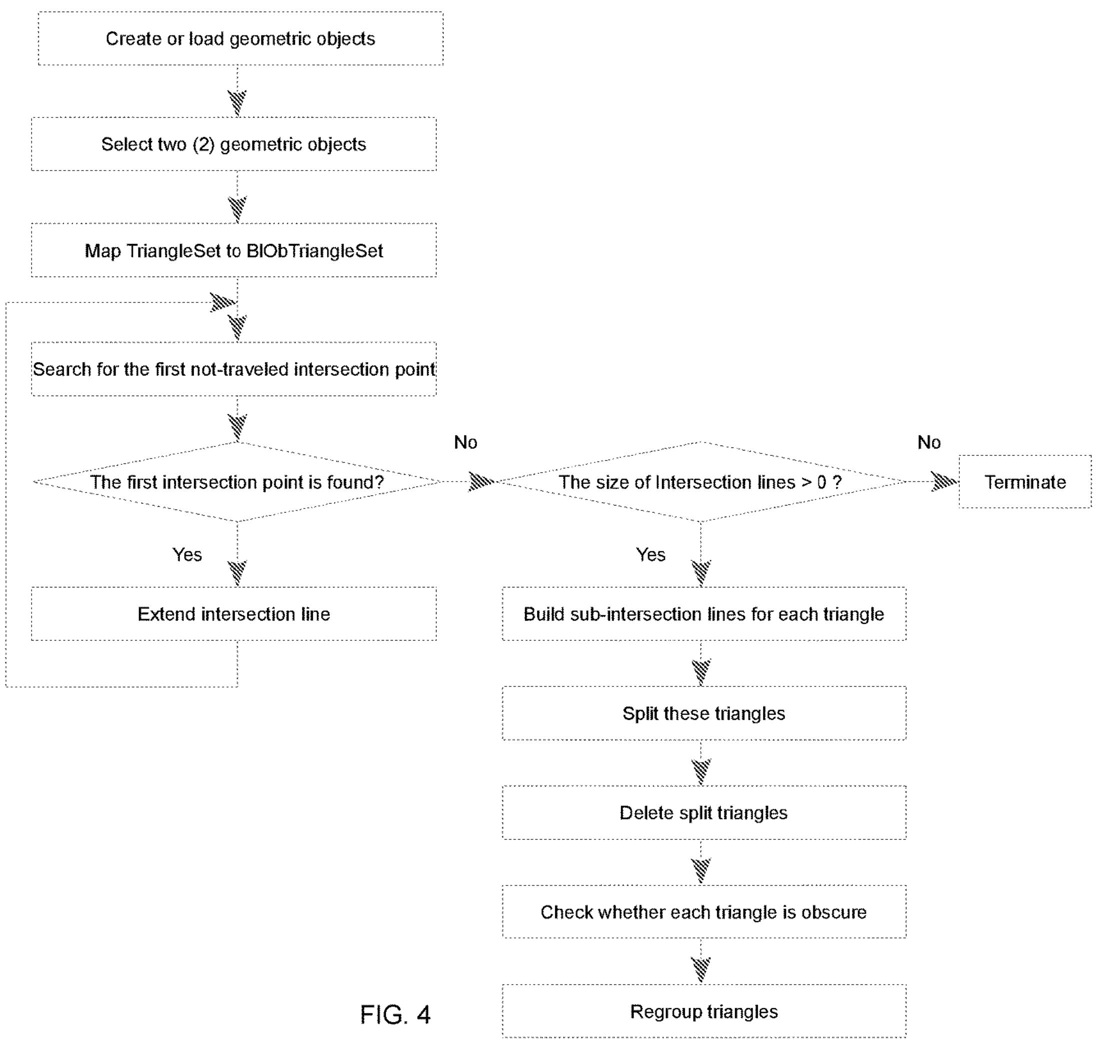


FIG. 4

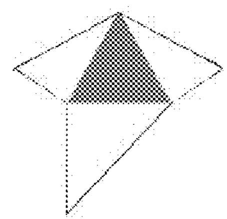


FIG. 5

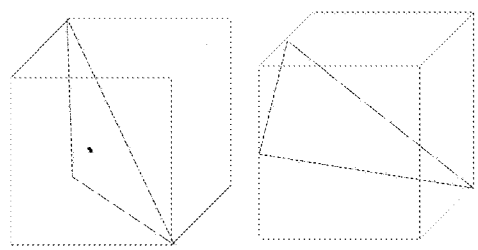


FIG. 6A

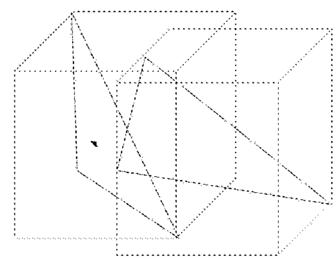


FIG. 6B

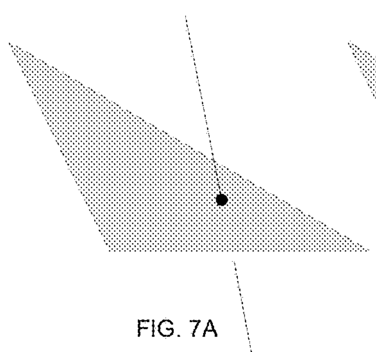


FIG. 7A

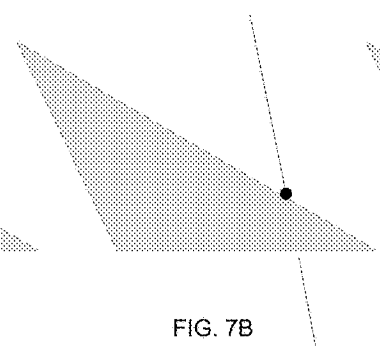


FIG. 7B

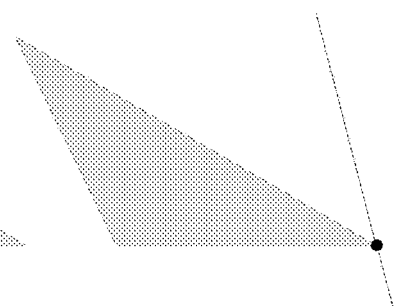


FIG. 7C

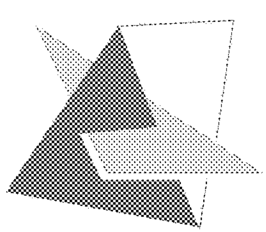


FIG. 8A

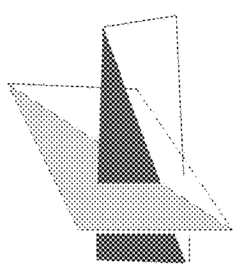


FIG. 8B

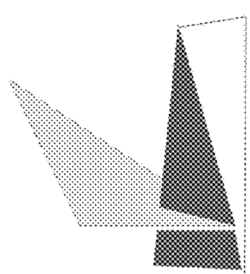


FIG. 8C

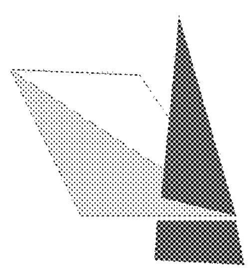


FIG. 8D

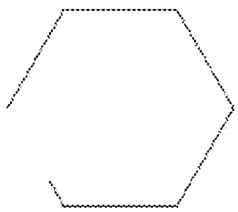


FIG. 9A

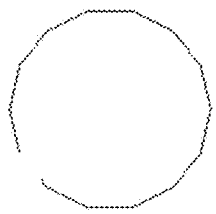


FIG. 9B

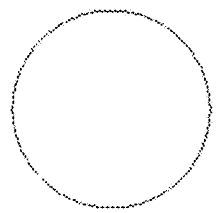


FIG. 9C

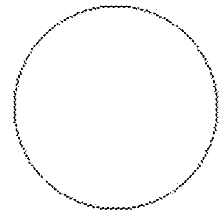


FIG. 9D

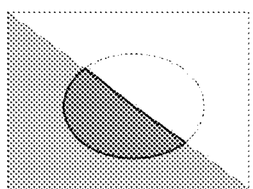


FIG. 10A

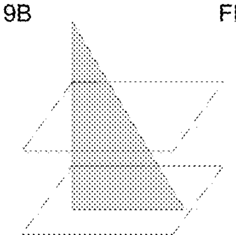


FIG. 10B

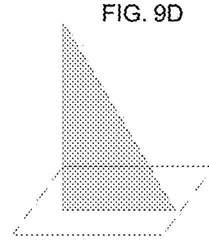


FIG. 10C

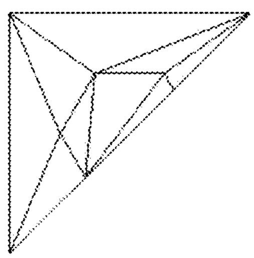


FIG. 11A

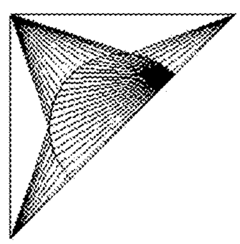


FIG. 11B

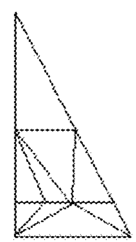


FIG. 11C

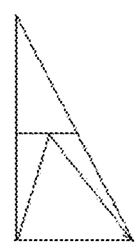


FIG. 11D

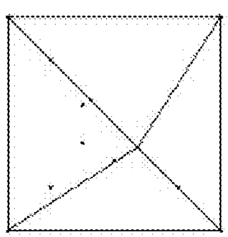


FIG. 12A

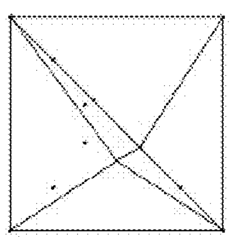


FIG. 12B

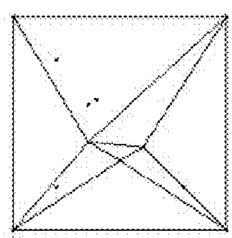


FIG. 12C

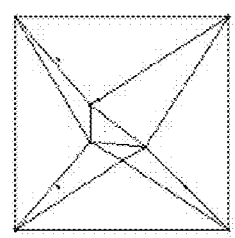


FIG. 12D

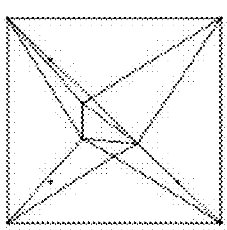


FIG. 12E

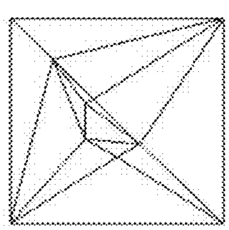


FIG. 12F

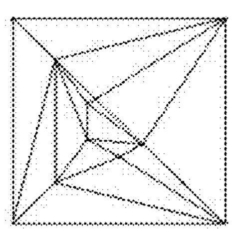


FIG. 12G

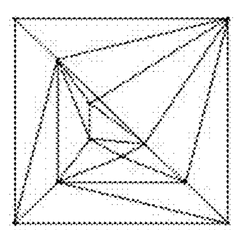


FIG. 12H

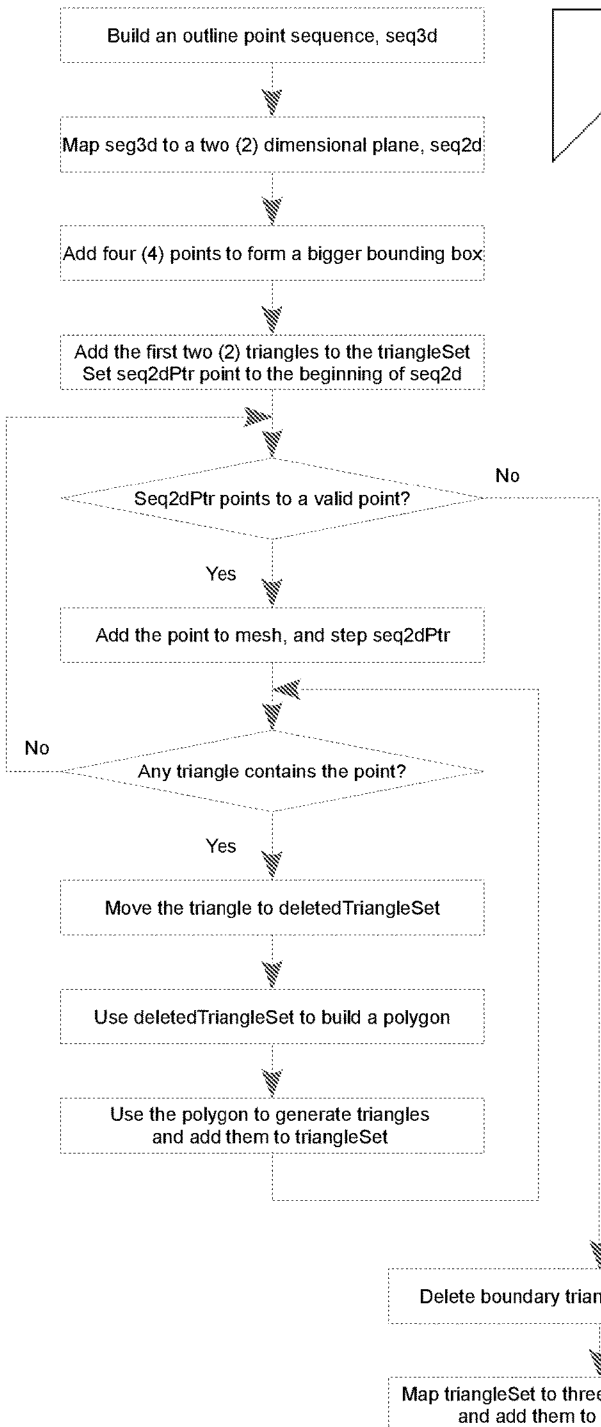


FIG. 13

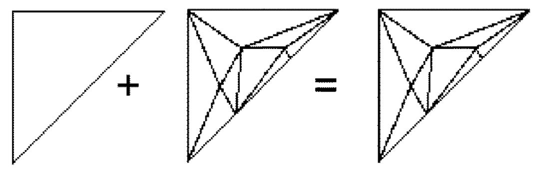


FIG. 14

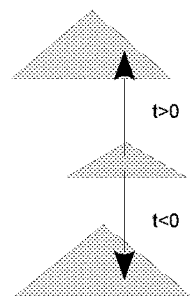


FIG. 15

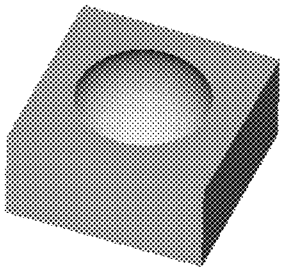


FIG. 16A

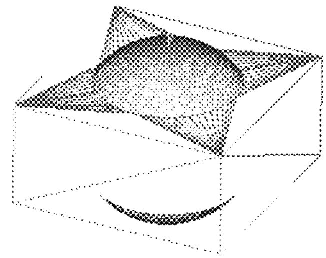


FIG. 16F

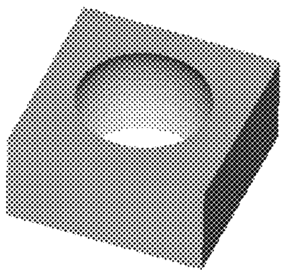


FIG. 16B

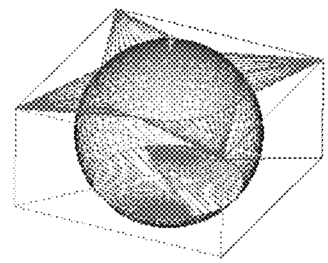


FIG. 16G

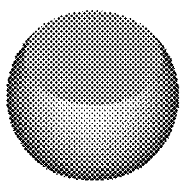


FIG. 16C

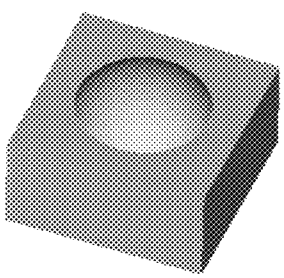


FIG. 16D

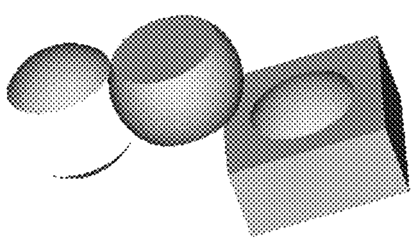


FIG. 16E

METHOD FOR IMMEDIATE BOOLEAN OPERATIONS USING GEOMETRIC FACETS

BACKGROUND

Field of the Invention

This invention provides an immediate Boolean operation method for building three (3) dimensional solid geometric models from primary geometric objects to Computer Aided Design, Computer Graphics, and Solid Modeling systems, which are widely used in product design, manufacturing, and simulation. Mechanic industry, culture and sports, everywhere there are geometric shapes, may have CAD/CG applications.

Related Art

Computer hardware is so highly developed that even an ordinary Personal Computer may be used to install and run a commercial CAD/CG system, which normally has Boolean operation functions including AND, OR, and NOT. PC components comprise input devices, such as a mouse and a keyboard, a main machine, a screen, and a printer. The software system contains geometric and non geometric functions. FIG. **1** shows the main PC components and FIGS. **2**A through **2**D depict a typical CAD/CG software system architecture.

Boolean operations provide a general process of building complex solid geometric objects from different geometric shapes, which include primary geometric objects, swept or extruded objects, to CAD/CG/Solid Modeling systems. Lee applied Boolean operations to divide surface [Lee U.S. Pat. No. 6,307,555].

Boolean operations may rely on Constructive Solid Geometry, CSG, to record primary geometry objects and operation sequence in a hierarchical way, which technically is easy to implement, whereas Boundary Representation, B-REP, is regarded as a more flexible way that supports more geometric object types like extended geometries [Gursoz, 1991].

This invention presents five (5) Boolean operation commands: combination, intersection, exclusion, difference, and division, which directly work on triangles decomposed from geometric facets used for rendering functions and do not require the data structure Constructive Solid Geometry or Boundary Representation. The data structures defined in this invention are a few of simple classes, the algorithms incorporated in this invention are concise and easy to implement, and the five (5) commands allow the user to create geometric models not only by selecting the types of geometric objects but also by defining their facets. FIGS. **3**A through **3**D present a box with **6** facets and a sphere with different facets make distinct results.

This invention presents different data structures and algorithms compared with CSG and B-REP, the algorithms include triangle-triangle intersections, building intersection lines, splitting each triangle with sub-intersection lines, and regrouping triangles to form Boolean operation results.

BRIEF SUMMARY OF THE INVENTION

This invention provides a set of data structures and algorithms for performing Boolean operations, which are used to build complex geometric models and work directly on triangles decomposed from geometric facets used as rendering data by computer hardware and rendering functions like OpenGL libraries. A geometric shape, for example, a sphere, a cone, a cylinder, a box, triangular facets, an extruded or swept object, is triangulated to form a set, noted as TriangleSet, for displaying. When two geometric objects are selected for performing a Boolean operation, neighboring triangles will be added to each triangle in TriangleSet to form another set, BlOpTriangleSet.

The second step of a Boolean operation this invention described is to search and build intersection lines between triangle sets. It starts with finding the first pair of intersecting triangles: this system builds an axis aligned minimum bounding box for each triangle and checks whether two bounding boxes overlap to decide if edge-triangle intersection needs to be calculated. Once the edge-triangle intersection point(s) falls inside a triangle, this system completes the searching task and stores the point data into an intersection line set.

To extend the current intersection line, this method traces neighboring triangles and calculates edge-triangle intersection points until the intersection line becomes closed or all triangles are traversed.

The third step of a Boolean operation this invention described is to split triangles. Each segment of the intersection lines references two (2) triangles, each of the triangles has at least one sub-intersection line that contains one or more segments, which divide a triangle into three (3) or more smaller triangles. After splitting the triangles, the original triangles are removed, and those smaller triangles are added to the BlOpTriangleSet.

The fourth step of a Boolean operation this invention described is to decide if each triangle is obscure or visible. If a triangle is enclosed by other triangles, it is obscure. A triangle is visible means it is outside another object.

The fifth step of a Boolean operation this invention described is to regroup the triangles: some of them have to be removed and some need to be put together, and there are five (5) cases for regrouping.

The final step of a Boolean operation this invention described is to map BlOpTriangleSet to TriangleSet.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** shows the main personal computer components, which generally contain a main machine, input devices including a mouse and a keyboard, a display, and a printer. A highly developed CAD/CG system can run on a PC machine.

FIGS. **2**A through **2**D describe a software architecture in which a CAD/CG/Solid Modeling system uses Boolean operations to build geometric models.

FIGS. **3**A through **3**D represent that different facets make various results even their original geometric object types and sizes are the same: the left side example has less facets and the right side has more facets. In these examples, Boolean Intersection operations work on a box and a sphere.

FIG. **4** is a flowchart for immediate Boolean operations using geometric facets.

FIG. **5** depicts that a triangle has three (3) neighbors. Given a triangle and its two vertices, there is one and only one neighboring triangle in solid models.

FIGS. **6**A and **6**B show two minimum bounding boxes do not overlap and two boxes overlap each other. Each triangle virtually has a minimum bounding box. If two boxes do not overlap, the triangles contained in the two boxes do not intersect. If the boxes overlap, edge-triangle intersection calculation is required.

FIGS. **7**A through **7**C depict three (3) edge-triangle intersection cases: an intersection point falls inside a triangle, an intersection point locates on an edge of a triangle, an intersection point is a vertex of a triangle.

FIGS. **8**A through **8**D show the searching candidate set, which allows the system to travel next triangle for extending intersection lines by conducting edge-triangle calculation. Triangles filled with colors are the last pair triangles that intersect each other, the triangles not filled are referenced by the member m_NeigTri of the data structure Triangle3dEx, which guides the system searching a minimum set of triangles when building intersection lines. The set contains one triangle, two triangles, or zero.

FIGS. **9**A through **9**D show four (4) examples of intersection lines. A box intersects a sphere, which has different facet numbers.

FIGS. **10**A through **10**C give three (3) examples of sub-intersection lines in darker color. FIG. **10**A has one (1) sub-intersection line, **10**B two (2), and **10**C one (1).

FIGS. **11**A through **11**D show four (4) examples that sub-intersection lines divided a triangle into a set of triangles.

FIGS. **12**A through **12**H show a Delaunay mesh sequence in which each intersection point is inserted into the mesh step by step.

FIG. **13** is the flowchart of Delaunay mesh modified Watson method that created the sequence of FIGS. **12**A through **12**H.

FIG. **14** shows that a triangle and its Delaunay mesh. The original triangle is removed and only the Delaunay mesh is reserved for late computations.

FIG. **15** shows t-Buffer where t may be negative and positive. If the size of negative t and positive t is balanced in t-Buffer, the triangle concerned is closed by another object and is obscure.

FIGS. **16**A through **16**E show five (5) examples of Boolean operations conducted with a box and a sphere. FIGS. **16**F and **16**G depict the internal mesh of two Boolean operation resultants: combination and exclusion.

DETAILED DESCRIPTION

This invention defines these data structures: Point3dEx, Triangle3dEx, and BlOpTriangle3dSet that inherit Point3d, Triangle 3d, Triangle3dSet storing facets for rendering geometric objects. When performing a Boolean operation, the system maps rendering facets to BlOpTriangle3dSet and all following processes focus on the members and attributes of BlOpTriangle3dSet. FIG. **4** is the flowchart describing the main procedure of Boolean operations conducted by the present invention. After a Boolean operation completed, the system maps the resultant stored in BlOpTriangle3dSet to rendering facets.

Geometric Facets for Rendering

CAD systems render facets to represent a geometric object, such as a sphere, a cone, a box, a cylinder, an extruded or swept object. A facet may compose three (3) or more points, and facets are usually decomposed into triangles for easy calculations. A box has six (6) facets decomposed into twelve (12) triangles. A sphere may have eighteen (18) facets, composing twenty four (24) triangles. A sphere may also be rendered using more than one thousand (1,000) facets and triangles. FIGS. **3**A through **3**D show a sphere rendered with different facets. This method uses Triangle3dSet to note triangle set data structure for

rendering a geometric object, it contains two (2) attributes: a three (3) dimensional point set and a triangle set, where Triangle3d references Point3d.

```
class Triangle3dSet
{
DataSet<Point3d> m_PointSet;
DataSet<Triangle3d> m_Tri angleSet;
};
class Triangle3d
{
Reference<Point3d> m_P0, m_P1, m_P2;
};
class Point3d
{
DataTypeI m_X, m_Y, m_Z;
};
```

Triangles for Boolean Operations

The Boolean Operation method described in this invention defined three (3) key classes: BlOpTriangleSet, Triangle3dEx, and Point3dEx.

```
class BlOpTriangleSet
{
  DataSet<Point3dEx>          m_PointSet;
  DataSet<Triangle3dEx>       m_TriangleSet;
};
class Point3dEx : Point3d
{
     DataTypeII      m_ID; // position and sequence index
     DataTypeIII     m_X, m_Y, m_Z; // DataType III
                     may be different from DataTypeI
};
class Triangle3dEx : Triangle3d
{
     DataTypeII      m_ID;
     Plane           m_Plane;
     DataTypeIV      m_Normal[3];
     Triangle3dEx*m_NeigTri[3]; // neighboring triangles
};
```

DataTypeII may be int, long, unsigned long, or other integer types. DataTypeIII is a floating point data type, such as float, double, even long double.

The class Triangle3dEx specifies each triangle may have 3 neighboring triangles, and every triangle is stored just one (1) copy in BlOpTriangleSet. Given the box example, the simplest way it has 12 triangles, even each of them has three (3) neighbors, BlOpTriangleSet still stores a total of 12 triangles.

Technically Triangle3d may have the attribute m_Normal. If DataTypeI and DataTypeIV are the same type, for example, double, the attribute m_Normal can be inherited.

Data Mapping

The process of mapping Triangle3dSet to BlOpTriangleSet copies point set and triangle set from rendering statue and fills default attributes. Data mapping contains the following procedures:

1) Copy points from Triangle3dSet to BlOpTriangleSet and ensure there are not identical points.

2) Copy triangles from Triangle3dSet to BlOpTriangleSet.

3) For each triangle in BlOpTriangleSet, set its neighboring triangles.

4) Calculate the normal and build the plane equation for each triangle in BlOpTriangleSet.

Remark 1: Given two (2) points a and b, if $|x_a-x_b|<\varepsilon$ and $|y_a-y_b|<\varepsilon$ and $|z_a-z_b<\varepsilon$, where ε is a positive float pointing number, for example 5.0e-16, then b is identical to a.

Remark 2: When mapping points from rendering data to BlOpTriangleSet, the system checks if there is an identical point in BlOpTriangleSet.

Remark 3: A triangle, which has three (3) points, defines a plane whose mathematical formula is ax+by+cz+d=0 and the class Plane internally records it as an array of four (4) numbers, such as double m_ABCD[4].

Remark 4: A triangle, if its three (3) points are not identical, always has a valid normal. Even it is related to m_ABCD, a separate copy makes things more clear and easy to handle later.

Remark 5: Every triangle has three (3) edges, when there are no duplicated points, it has three (3) neighboring triangles in solid models. FIG. **5** shows an example: a triangle filled with dark color and its three (3) neighbors.

The First Intersection Point

Every triangle has three (3) vertices, which define a minimum bounding box. This method adopted the concept of axis aligned minimum bounding box.

Given a pair of triangles, if their bounding boxes do not overlap, the two triangles have no intersection point; otherwise, this method carries out edge-triangle intersection calculations.

If an edge of a triangle T_a intersects with a plane defined by a triangle T_b and the intersection point pet falls inside T_b, then pet is the first intersection point. If pet is outside of T_b, then switch the triangle position in the pair, (T_a, T_b) changed to (T_b, T_a), and conduct edge-triangle intersection calculations.

Given the i-th edge of a triangle T_a, $i\in[0, 2]$, its formula is: $p=p_i+t^*(p_{(i+1)\%3}-p_i)$, and the plane defined by the triangle T_b, its formula is: ax+by+cz+d=0. Hereinafter the symbol % expresses the modulo operator. If the two formulas have a solution, the edge intersects with the plane. If the edge-plane intersection point falls inside the triangle T_b, then the point is the edge-triangle intersection point.

Extending an Intersection Line

This method defines a data structure for recording an intersection point as PntEgTri:

```
class PntEgTri
{
        Triangle3dEx     *m_Tri0, *m_Tri1;
        DataTypeII       m_EdgeIndex;
        DataTypeII       m_PointPosi;
        Point3dEx        m_Point;
        Point3dEx        *m_PntGlobalIndexA, *m_PntGlobalIndexB;
};
```

According to the location of an intersection point on a triangle, a PntEgTri, simply said pet, can be classified into three (3) categories shown in FIGS. **7**A through **7**C.

1) The most popular case is edge-triangle intersection, pet locates on an edge of triangle T_a and inside triangle T_b.
2) Edge-edge intersection, pet locates on an edge of triangle T_a and on an edge of triangle T_b.
3) Edge-vertex intersection, pet locates on an edge of triangle T_a and on a vertex of triangle T_b.

To extend an intersection line, this method catches next neighboring triangle(s) and checks edge-triangle intersection until the intersection line gets closed or all triangles are traversed.

Sub-Intersection Line

An intersection line passes through a set of triangles and divides each triangle into multi partitions. The segments of an intersection line inside a triangle make up a sub-intersection line. FIGS. **10**A through **10**C show three examples in which the dark lines are sub-intersection lines. In practice, a triangle may have zero (0), one (1), two (2), or three (3) sub-intersection lines.

The following algorithm shows how to get a valid reference to a triangle that has at least one sub-intersection line:

for each intersection line
for each intersection point, get the triangle references: (m_Tri0, m_Tri1)
  for each triangle of the triangle pair, if it is not split
    for each intersection line
        search and build a sub-intersection line

Given a valid triangle and an intersection line, to decide if a pet belongs to the sub-intersection line of the triangle, this method checks whether

1) pet is on an edge of the triangle,
2) or pet is inside the triangle,
3) or pet equals a vertex of the triangle.

Splitting a Triangle

Given a set of sub-intersection lines, to split a triangle, this method

1) Removes duplicated pets. If neighboring pets are identical, this method reserves just one copy.
2) Identifies the position of end pets: checks each pet locates on which edge of the triangle.
3) Splits the upper partition, lower partition, and middle partition of the triangle where applicable.

Given a set of points on a plane that represents a partition of a triangle, to decompose the plane into a group of triangles, this invention modified Delaunay 2D mesh Watson method, which is published in 1981 [Watson, 1981].

A Delaunay 2D mesh has three (3) data set: triangle set that holds the generated triangles, deleted triangle set that stores just deleted triangles, and polygon that records the outline of deleted triangle set.

The modified Delaunay 2D mesh method contains the following steps:

1) Build an outline point sequence that links sub-intersection lines and vertices of the triangle where applicable;
2) Map the three (3) dimensional point sequence to two (2) dimensional points according to the aspect of the plane;
3) Add four (4) points to form a bigger bounding box that encloses all the two (2) dimensional points;
4) Assume that one dialog line of the bounding box splits the box into two (2) triangles and add them into the triangle set;
5) Insert every point except that bounding ones into the triangle set.
   a) For each point, check every triangle in the triangle set whether its circumcircle contains the point or the last segment of the outline passes through the triangle. If the condition is true, erase it from the triangle set and add it to the deleted triangle set.

b) Use the deleted triangle set to extend polygon and clear the deleted triangle set immediately.

c) Use the polygon to generate triangles and add them to the triangle set.

6) Delete boundary triangles from the triangle set.

7) Map the triangle set to three (3) dimensional triangles and add them to BlOpTriangleSet.

FIGS. **12**A through **12**H show a Delaunay 2D mesh sequence.

Deleting Split Triangles

In the above step, a split triangle got a mark. After all triangles have been traversed, this method deletes the marked triangles. FIG. **14** shows a deletion result.

Obscure Facets

Given two sets of triangles A and B, if A bounds a triangle of B, T_b, then T_b is obscure; if B bounds a triangle of A, T_a, then T_a is obscure.

To check whether a triangle T is bounded by an object O, this invention uses the following steps.

(1) Calculate the centred, c, of the triangle, T.

(2) Build a line I:p=c+t*N, which passes through the centroid and along the normal N of the triangle T.

(3) For each triangle T_0 of the object O, calculate line-plane intersection point. If there is a valid intersection point pet that falls inside the triangle T_0, then calculate t that is determined by centroid c and the pet, and add t to a depth buffer, buffer T.

(4) Check the size of negative t and positive t stored in buffer T. If the two sizes are equal, then the triangle T is bounded and obscure.

Regrouping the Facets

This invention presents five (5) kinds of Boolean operations: combination, intersection, exclusion, difference, and division, each of them has a different regrouping procedure. The combination operation, logically it is OR, combines two solid geometric objects and generates a new object, which normally discards obscure partitions and reserves visible ones viewing from outside, has the following procedure.

1) Delete obscure triangles of object A;

2) Delete obscure triangles of object B;

3) Merge the triangles of object A and B.

The intersection operation, logically it is AND, which creates a solid geometric object using public section(s) of two geometric objects and discards any partitions of A and B outside the shared public section(s), has the following procedure.

1) Delete NOT obscure triangles of object A;

2) Delete NOT obscure triangles of object B;

3) Merge the triangles of object A and B.

The exclusion operation, which builds a solid geometric object by removing public section(s) of two geometric objects and keeps not shared partitions, has the following procedure.

1) Copy object A's obscure triangles to a buffer, buffer A;

2) Delete obscure triangles from object A;

3) Copy object B's obscure triangles to object A;

4) Delete obscure triangles from object B;

5) Copy the triangles in buffer A to object B;

6) Reverse the normal of every obscure triangle of A and B;

7) Merge the triangles of the two objects.

The difference operation, which cuts geometric object A with another object B by removing any partitions of A inside B, has the following procedure.

1) Delete obscure triangles of object A;

2) Delete NOT obscure triangles of object B;

3) Reverse the normal of every triangle of object B;

4) Merge triangles of object A and B.

The division operation, which divides two solid geometric object A and B into three (3) objects, public section(s) of the two geometric objects, the NOT shared partitions of A and partitions of B, has the following procedure.

1) Copy object A's obscure triangles to a buffer, buffer A;

2) Copy object B's obscure triangles to buffer A;

3) Copy object A's obscure triangles to another buffer, buffer B;

4) Delete object A's obscure triangles;

5) Copy object B's obscure triangles to object A;

6) Delete object B's obscure triangles;

7) Copy object A's obscure triangles stored in buffer A to object B;

8) Reverse the normal of every obscure triangles of A and B.

Mapping to Rendering Facets

Once a Boolean operation is finished, this method maps BlOpTriangleSet to rendering triangles.

1) Each Point3dEx of BlOpTriangleSet is mapped to a Point3d of TriangleSet;

2) Each Triangle3dEx of BlOpTriangleSet is mapped to a Triangle3d of TriangleSet.

U.S. PATENT DOCUMENTS

U.S. Pat. No. 6,307,555, 10-2001, Lee; Eugene T. Y. 345/421.

OTHER PUBLICATIONS

"Boolean Set Operations on Non-Manifold bounding Representation Objects", E. Gursoz et al., Computer-Aided Design 23 (1991) January/February No. 1 London, GB.

"Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopest", D. F. Watson, The Computer Journal 24 (2) 1981.

What is claimed:

1. A method that performs immediate Boolean operations using geometric facets of geometric objects implemented in a computer system and operating with a computer, the method comprising:

mapping rendering facets to extended triangles that contain neighbors;

building intersection lines starting with and ending with searching for the first pair of triangles that hold a start point of an intersection line by detecting whether two minimum bounding boxes overlap and performing edge-triangle intersection calculations for locating an intersection point, then searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed;

splitting each triangle through which an intersection line passes using modified Watson method, wherein the modified Watson method includes removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion, and a middle portion;

checking each triangle whether it is obscure or visible for Boolean operations;

regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets; and

mapping extended triangles to rendering facets.

2. The method of claim **1** wherein any Boolean operations that use rendering facets of the geometric objects to create new geometric objects, including combination, intersection, exclusion, difference, and division, map rendering facets to extended triangles, build intersection lines, split each triangle through which an intersection line passes, check each triangle whether it is obscure or visible, regroup facets to form new extended triangle sets, and map extended triangles to rendering facets without the data structure Constructive Solid Geometry and Boundary Representation.

3. The method of claim **1** wherein any Boolean operations that use rendering facets of the geometric objects to create new geometric objects, including combination, intersection, exclusion, difference, and division, map rendering facets to extended triangles, build intersection lines, split each triangle through which an intersection line passes, check each triangle whether it is obscure or visible, regroup facets to form new extended triangle sets, and map extended triangles to rendering facets with the data structure CSG or B-REP.

4. The method of claim **1** wherein searching for the first pair of triangles that hold a start point of an intersection line and searching neighboring triangles of the last triangle pair that hold the last intersection point composed the procedure for building an intersection line that usually repeats more than one time when building intersection lines use the minimum bounding boxes to detect whether two triangles do not overlap and carry out edge-triangle intersection calculations comprising the steps ensuring that the intersection points are exact and the intersection lines are not approximate curves: building the formula $p=p_i+t^*(p_{(i+1)\%3}-p_i)$ for expressing the i-th edge of the triangle T_a that is one triangle inside the triangle pair, building the formula ax+by+cz+d=0 for recording the plane defined by the triangle T_b that is another triangle inside the triangle pair, and getting the solution of the two linear formulas.

5. The method of claim **1** wherein searching for the first pair of triangles and searching neighboring triangles calculate edge-triangle intersection and employee neighboring triangles ensuring that direct calculation of edge-edge intersection is replaced by verifying whether a point is on an edge of a triangle.

6. The method of claim **1** wherein splitting each triangle projects the three (3) dimensional triangle and all its sub-intersection lines onto a two (2) dimensional plane and builds Delaunay 2D mesh with modified Watson method that defines a triangle set, a deleted triangle set, and a polygon, dividing the triangle into different partitions even when the sub-intersection lines are not convex, comprising of: building an outline point sequence that links sub-intersection lines and vertices of the triangle where applicable; mapping the three (3) dimensional point sequence to two (2) dimensional points according to the aspect of the plane defined by the triangle; adding four (4) points to form a bigger bounding box that encloses all the two (2) dimensional points; assuming that one dialog line of the bounding box splits the box into two (2) triangles and adding them into the triangle set; inserting every point except bounding ones into the triangle set with the steps: for each point, checking every triangle in the triangle set whether its circumcircle contains the point or the last segment of the outline passes through the triangle, and when the condition is met, erasing it from the triangle set and adding it to the deleted triangle set, using the deleted triangle set to extend the polygon and clearing the deleted triangle set immediately, and using the polygon to generate triangles and adding them to the triangle set; deleting boundary triangles from the triangle set; and mapping the triangle set to three dimensional triangles and adding them to BlOpTriangleSet.

7. The method of claim **1** wherein triangles are classified as either visible, in which a visible triangle is not enclosed by a geometric object, or obscure in which an obscure triangle is enclosed by a geometric object.

8. The method of claim **1** wherein checking each triangle whether it is obscure or visible utilizes t-Buffer further comprising:

calculating the centroid c of triangle T_a that belongs to geometric object A;

building a line I: p=c+t*N passing through the centroid c and along the normal of T_a;

for each triangle T_b of object B, checking whether I intersects with T_b at an interior point and adding t to a depth buffer, t-Buffer, and setting T to be obscure when the size of negative t equals to that of positive tin t-Buffer.

9. The method of claim **1** wherein a Boolean operation that is a combination, an intersection, an exclusion, a difference, or a division, regroups facets for constructing its operational result using one or more steps of: deleting obscure or visible triangles of an object, copying obscure triangles of an object to a buffer or copying triangles from a buffer to an object, reversing the normal of each triangle of an object, and merging the triangles of the objects to form new extended triangle sets.

10. The method of claim **1** wherein the extended triangles are directly mapped to rendering facets for being displayed and providing data to next Boolean operations.

11. A computer system consisting of hardware and software that performs immediate Boolean operations using rendering facets of geometric objects, the system comprising:

a computer with input devices for entering data and commands, and a display device showing user interface, geometric objects, and additional data, having a medium storing geometric data and instructions that make up of a software system, or having a microchip or integrated circuit embedding partially or totally the instructions, and a processor that executes the steps of:

creating, modifying or loading primary geometric objects including swept and extruded ones and relocating them at different positions or orientations with input devices of the computer;

selecting two of the geometric objects;

mapping rendering facets to extended triangles that contain neighbors;

building intersection lines starting with and ending with searching for the first pair of triangles that hold a start point of an intersection line by detecting whether two minimum bounding boxes overlap and by performing edge-triangle intersection calculations for locating an intersection point, then searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed;

splitting each triangle through which an intersection line passes using modified Watson method, wherein the modified Watson method includes removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion, and a middle portion;

checking each triangle whether it is obscure or visible for Boolean operations;

regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets; and

mapping extended triangles to rendering facets.

12. The system of claim **11** wherein any Boolean operations that use rendering facets of the geometric objects to create new geometric objects, including combination, intersection, exclusion, difference, and division, map rendering facets to extended triangles, build intersection lines, split each triangle through which an intersection line passes, check each triangle whether it is obscure or visible, regroup facets to form new extended triangle sets, and map extended triangles to rendering facets without the data structure Constructive Solid Geometry indicating the acronym CSG, and Boundary Representation indicating the acronym B-REP.

13. The system of claim **11** wherein any Boolean operations that use rendering facets of the geometric objects to create new geometric objects, including combination, intersection, exclusion, difference, and division, map rendering facets to extended triangles, build intersection lines, split each triangle through which an intersection line passes, check each triangle whether it is obscure or visible, regroup facets to form new extended triangle sets, and map extended triangles to rendering facets without with the data structure CSG or B-REP.

14. The system of claim **11** wherein searching for the first pair of triangles that hold a start point of an intersection line and searching neighboring triangles of the last triangle pair that hold the last intersection point composed the procedure for building an intersection line that usually repeats more than one time when building intersection lines use the minimum bounding boxes to detect whether two triangles do not overlap and carry out edge-triangle intersection calculations comprising the steps ensuring that the intersection points are exact and the intersection lines are not approximate curves: building the formula $p=p_i+t^*(p_{(i+1)\%3}-p_i)$ for expressing the i-th edge of the triangle T_a that is one triangle inside the triangle pair, building the formula ax+by+cz+d=0 for recording the plane defined by the triangle T_b that is another triangle inside the triangle pair, and getting the solution of the two linear formulas.

15. The system of claim **11** wherein searching for the first pair of triangles and searching neighboring triangles calculate edge-triangle intersection and employee neighboring triangles ensuring that direct calculation of edge-edge intersection is replaced by verifying whether a point is on an edge of a triangle.

16. The system of claim **11** wherein splitting each triangle builds Delaunay 2D mesh with modified Watson method that defines a triangle set, a deleted triangle set, and a polygon, dividing the triangle into different partitions even when the sub-intersection lines are not convex, comprising of: building an outline point sequence that links sub-intersection lines and vertices of the triangle where applicable; mapping the three (3) dimensional point sequence to two (2) dimensional points according to the aspect of the plane defined by the triangle; adding four (4) points to form a bigger bounding box that encloses all the two (2) dimensional points; assuming that one dialog line of the bounding box splits the box into two (2) triangles and adding them into the triangle set; inserting every point except bounding ones into the triangle set with the steps: for each point, checking every triangle in the triangle set whether its circumcircle contains the point or the last segment of the outline passes through the triangle, and when the condition is met, erasing it from the triangle set and adding it to the deleted triangle set, using the deleted triangle set to extend the polygon and clearing the deleted triangle set immediately, using the polygon to generate triangles and adding them to the triangle set; deleting boundary triangles from the triangle set; mapping the triangle set to three dimensional triangles and adding them to BIOpTriangleSet.

17. The system of claim **11** wherein triangles are classified as either visible, in which a visible triangle is not enclosed by a geometric object, or obscure in which an obscure triangle is enclosed by a geometric object.

18. The system of claim **11** wherein checking each triangle whether it is obscure or visible utilizes t-Buffer further comprising:

calculating the centroid c of triangle T_a that belongs to geometric object A;

building a line I: p=c+t*N passing through the centroid c and along the normal of T_a;

for each triangle T_b of object B, checking whether I intersects with T_b at an interior point and adding t to a depth buffer, t-Buffer; and setting T to be obscure when the size of negative t equals to that of positive tin t-Buffer.

19. The system of claim **11** wherein a Boolean operation that is a combination, an intersection, an exclusion, a difference, or a division, regroups facets for constructing its operational result using one or more steps of: deleting obscure or visible triangles of an object, copying obscure triangles of an object to a buffer or copying triangles from a buffer to an object, reversing the normal of each triangle of an object, and merging the triangles of the objects.

20. The system of claim **11** wherein the extended triangles are directly mapped to rendering facets for being displayed and providing data to next Boolean operations.

* * * * *

RUDY KIM (CA SBN 199426)
RudyKim@mofo.com
ROMAN A. SWOOPES (CA SBN 274167)
RSwoopes@mofo.com
MORRISON & FOERSTER LLP
755 Page Mill Road
Palo Alto, California 94304-1018
Telephone: 650.813.5600
Facsimile: 650.494.0792

Attorneys for Defendant
AUTODESK, INC.

UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION

NATURE SIMULATION SYSTEMS INC., Plaintiff, v. AUTODESK, INC., Defendant.	Case No. 3:19-cv-03192-SK **DECLARATION OF DANIEL G. ALIAGA IN SUPPORT OF AUTODESK'S CLAIM CONSTRUCTION POSITIONS**

I, Daniel G. Aliaga, declare as follows:

1. I submit this declaration in support of Autodesk's claim construction proposals for U.S. Patent Nos. 10,120,961 (the "'961 patent") and 10,109,105 (the "'105 patent"). I have personal knowledge of the facts set forth in this declaration and, if called upon as a witness, I could and would testify to such facts under oath.

I. SUMMARY OF OPINIONS

2. It is my opinion that to a person of ordinary skill in the art, the following terms do not and did not have an ordinary and customary meaning, including at the time of the alleged invention, and are indefinite:

- "modified Watson method" (see pages 4-9);
- "searching neighboring triangles of the last triangle pair that holds the last intersection point" (see pages 9-13);
- "regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets" (see pages 13-15);
- "regular points" (see pages 17-18);
- "BIOpTriangleSet," (see pages 18-20); and
- "according to m_ID of the member m_Points of each triangle, deciding whether it is a boundary triangle" (see pages 20-21).

3. Furthermore, it is my opinion that to a person of ordinary skill in the art, the following term has and had an ordinary and customary meaning, including at the time of the alleged invention:

- "surface trimming" (see pages 15-16).

II. QUALIFICATIONS

4. My curriculum vitae, attached as Exhibit A, provides an accurate identification of my relevant background and experience.

5. I graduated *magna cum laude* from Brown University in 1991, receiving my B.Sc.

in Computer Science and writing an Honor's Thesis related to computer graphics. I then proceeded to study computer science, focusing on computer graphics, at the University of North Carolina at Chapel Hill, where I received my M.S. in Computer Science in 1993 and my Ph.D. in 1999.

6. During my educational formation, I worked for several companies, including IBM, Silicon Graphics, Siemens, and Division Inc. After receiving my Ph.D., I worked as a full-time graphics and imaging researcher at Lucent Technologies Bell Laboratories until December 2002. Subsequently, I obtained a full time research position at Princeton University and then began my academic career at Purdue University in August 2003. I have been at Purdue University ever since, and I am currently an Associate Professor of Computer Science. In addition, I have been a visiting professor in the Department of Computer Science at ETH Zurich, in the Department of Architecture at ETH Zurich, in the Department of Computer Science at King Abdullah University of Science and Technology in Saudi Arabia, and at INRIA (French Computer Science/Computer Graphics Research Institute). Furthermore, I have roles in several startups, and I am a named inventor on eight patents to date. I have also received an Organization for Economic Co-operation and Development (OECD) Fellowship, a Discovery Park Faculty Research Fellowship, and Fulbright Scholar Award.

7. My first computer graphics publication was in 1990, and I have since published over 120 peer-reviewed publications in top journals and conferences.

8. I have chaired and served on numerous ACM and IEEE conference and workshop committees, including being a member of more than 70 program committees, a conference chair, a papers chair, an invited speaker, and an invited panelist. I also participate in several ongoing international multi-disciplinary collaborations (i.e., with world experts in computer science, photogrammetry, urban planning, architecture, meteorology, atmospheric sciences, earth sciences, traffic engineering, and more) and have given over 50 invited talks and presentations (including in the United States, Brazil, Colombia, Ecuador, France, Japan, Korea, Peru, Qatar, Sweden, and Switzerland). Moreover, I have served on several National Science Foundation ("NSF") panels and on the editorial board of Computer Graphics Forum and of Graphical Models. I am also a

Case: 20-2251 Document: 20 Page: 54 Filed: 03/12/2021

member of ACM SIGGRAPH.

9. My research has been wholly or partially funded by over $7 million in funds from NSF, Metropolitan Transportation Commission (of the State of California), Microsoft Research, Google, IARPA, Internet2, and Adobe Inc.

10. In my career in academia, I have been performing research, publishing papers, and producing software primarily in the field of computer graphics, but also in visualization and computer vision. In addition, I am the instructor for undergraduate and graduate computer graphics courses covering geometric operations, geometric modeling, triangulation, rendering, lighting, and other topics. The major part of my computer graphics work has been developing 3D modeling and rendering algorithms, including methods for geometric modeling and 3D reconstruction of objects and of scenes. All of these techniques employ graphics software and hardware, and many benefit from modern GPUs and GPU programming using OpenGL, GLSL, CUDA, Qt, and other libraries.

III. MATERIALS REVIEWED

11. I have reviewed the claims and specifications of the ’961 patent and the ’105 patent; their prosecution histories; D. F. Watson, “Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes,” The Computer Journal 24 (2) 1981, the Joint Claim Construction and Prehearing Statement (and the materials cited therein), and Nature Simulation Systems’ claim construction brief in preparing the opinions I present in this declaration.

IV. RELEVANT LEGAL STANDARDS

12. I understand that a court construing a patent claim seeks to give a claim the meaning it would have to a person of ordinary skill in the art (“POSITA”) at the time of the invention. I understand that claim construction requires consideration of the words of the claims themselves, the remainder of the specification, the prosecution history, and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art. I also understand that the specification may reveal a special definition given to a claim term by the patentee that differs from the meaning it would otherwise possess and that in such cases, the

inventor's lexicography governs.

13. I understand that the claims of a patent, viewed in the context of the specification and prosecution history, must inform a person of ordinary skill in the art at the time of the claimed subject matter as to the scope of the subject matter with reasonable certainty. I understand that a patent claim is invalid due to indefiniteness when it fails to inform with reasonable certainty those skilled in the art about the scope of the claimed subject matter when read in light of the specification and the prosecution history. I understand that while absolute precision is unattainable, the definiteness requirement nonetheless mandates clarity.

V. LEVEL OF SKILL IN THE ART

14. I believe that a person skilled in the art of the '961 patent and the '105 patent would have had at least a master's degree in computer science or a related field, or a bachelor's degree in computer science or a related field plus two years of relevant experience, with experience in computer graphics, computer-aided design, solid modeling, or geometric modeling. Based on my experience, I understand how a POSITA would have understood and used the terminology in the field of the '961 and '105 patents and in the relevant art at the time of the asserted patents' filings.

VI. OPINIONS RELATING TO CLAIM TERMS

A. "modified Watson method" ('961 and '105 patents)

NSS	Autodesk
[ordinary meaning]	[indefinite]

15. I understand that Nature Simulation Systems (NSS) has argued that the term "modified Watson method" has a plain and ordinary meaning and requires no construction. I disagree with this position because "modified Watson method" is not a known term of art. To be clear, I am familiar with the Delaunay method, which is a known method of triangulation that is mentioned in the patents. (*See, e.g*., '961 patent at 6:64-66; '105 patent, 6:42-44.) And I am also aware of the "Watson" algorithm for computing a Delaunay triangulation that is described in a 1981 paper by D.F. Watson cited in the patents. (*Id.*) However, in my professional experience, "modified Watson method" does not have a standardized meaning. It does not have and did not

have an ordinary and customary meaning, including at the time of the alleged invention, to a POSITA.

16. Since "modified Watson method" does not have a customary meaning to a POSITA, I searched for its meaning in the '961 and the '105 patents themselves. While the term is mentioned seven times in the patents, it is not clearly defined anywhere. Claim 1 of both patents states that "the modified Watson method includes removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion, and a middle portion," but this language does not resolve the confusion. For example, the patent does not describe how to identify "duplicate intersection points." Moreover, the phrase "splitting portion of each triangle including an upper portion, a lower portion, and a middle portion" is ambiguous. It could refer to (a) splitting an upper portion, splitting a lower portion, and splitting a middle portion of each triangle; (b) splitting one portion of each triangle, if that triangle has an upper portion, a lower portion, and a middle portion; (c) splitting each triangle into an upper portion, a lower portion, and a middle portion; or (d) something else. It is also unclear why a triangle split by a line would have three portions. Also, neither Delaunay triangulation nor Watson's algorithm deal with intersection points, so it would not be clear how intersection points fit into the "modified Watson method."

17. It is not even clear from the patents what the claimed "Watson method" is—i.e., which specific steps from Watson's algorithm are incorporated into the patents, and which are not—so it is not clear what specific steps are being modified or added. Figure 13 of the patents provides no clarification. The patents state that "FIG. 13 is the flowchart of Delaunay mesh modified Watson method that created the sequence of FIGS. 12A through 12H." ('961 patent, 3:39-41; '105 patent, 3:26-28.) Figure 13 of the '961 patent is shown below:

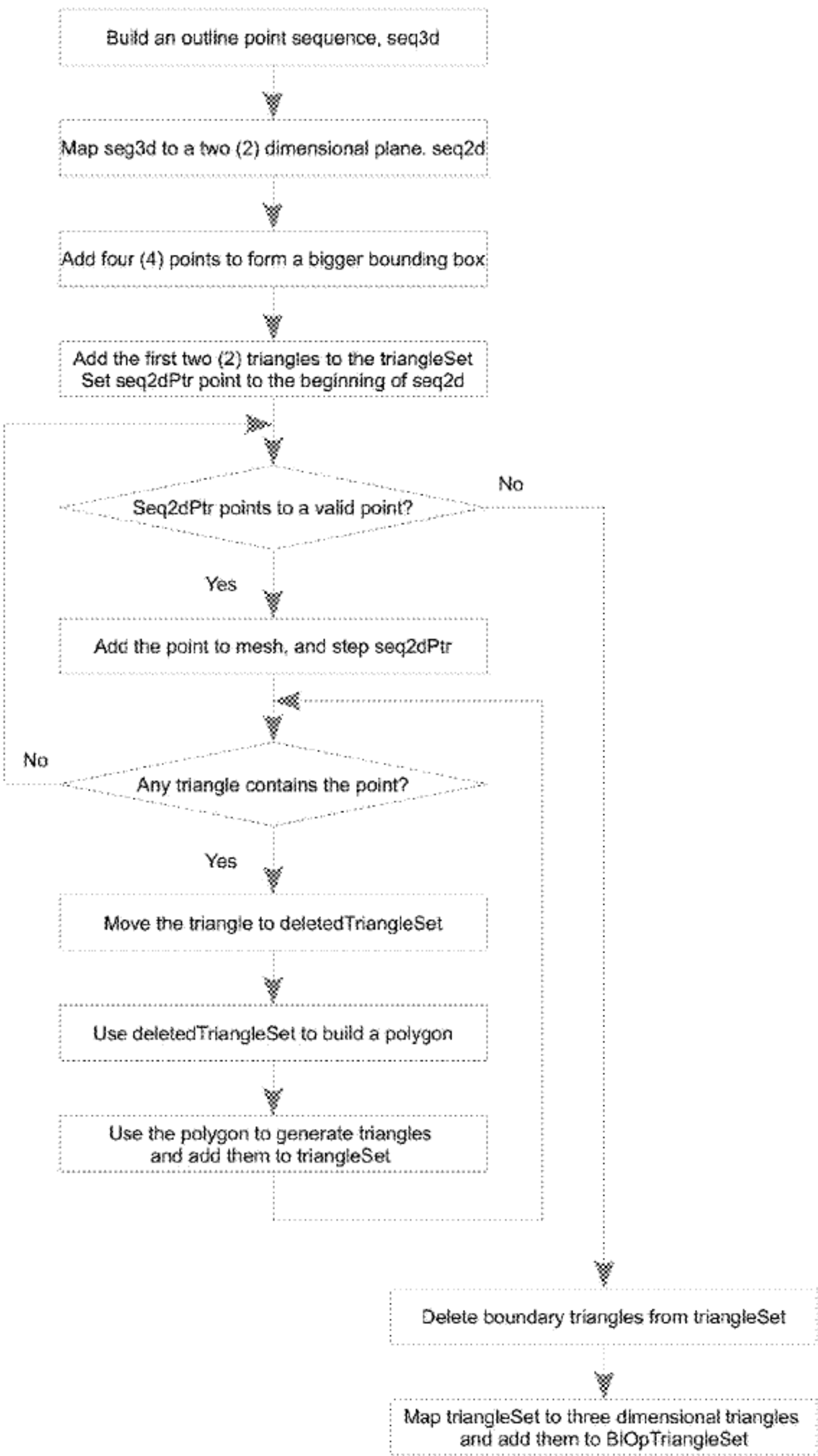


FIG. 13 Prior Art except the first two (2) steps , the last one, and the condition Any triangle contains the point.

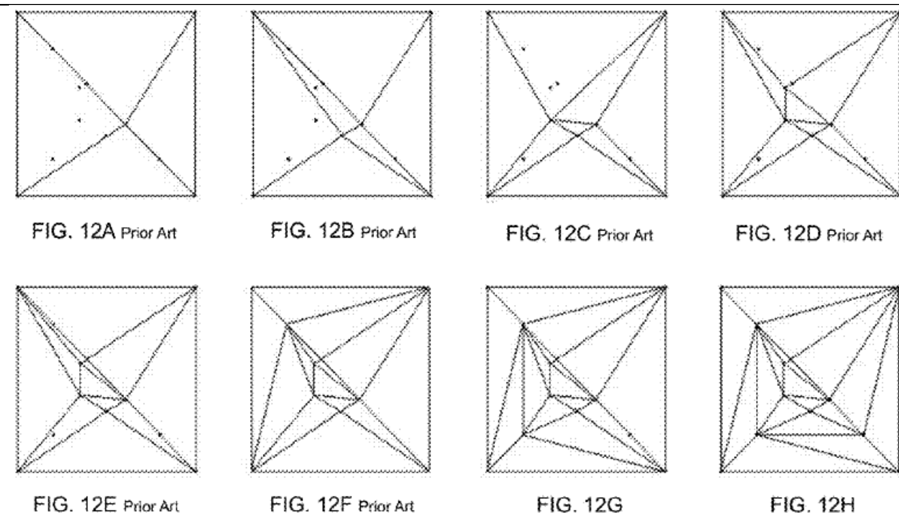
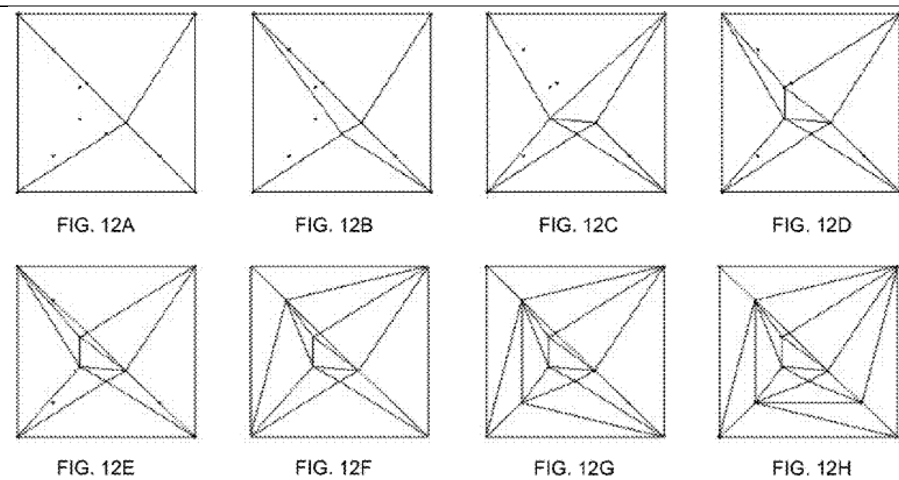
The patents also contain a block of text that appears to roughly correspond to the flowchart in Figure 13. (*See* '961 patent, 7:2-32 ("The modified Delaunay 2D mesh method contains the following steps….); '105 patent, 6:48-7:9 (same).) The label at the bottom of Figure 13 of the '961 patent specifies which steps are prior art and which are not: "Prior art except for the first two (2) steps, the last one, and the condition Any triangle contains the point."[1] But Figure 13 and the accompanying text are inconsistent with the claim language "the modified Watson method includes removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion, and a middle portion." This claim language does not appear in Figure 13. Moreover, the claim language states that the modified Watson method "includes" the aforementioned recited steps, which do not appear in Figure 13. In addition, the claim language does not specify which steps from Watson's algorithm are incorporated into the claim.

18. I understand that in its opening claim construction brief, NSS apparently argued that "modified Watson method" has a definite meaning because claim 1's condition of "splitting each triangle through which an intersection line passes" sets it apart from the prior art, where "Watson defined the splitting condition as the triangle's circumcircle containing the point." (ECF No. 35 at 8.). The relevant specification passage cited by NSS states: "For each point, check every triangle in the triangle set whether its circumcircle contains the point or the last segment passes through the triangle." ('961 patent, 7:19-21.) If NSS is suggesting that the very condition of splitting triangles on the intersection line is what "modifies" the Watson method, a POSITA would not have understood the claimed "modification" to eliminate the use of circumcircles from the "Watson method" of claim 1. Moreover, NSS' argument does not address the fact that the patent fails to make clear how the triangles on the intersection line are split according to the "modified Watson method" or what steps one must follow in order to perform such splitting. A POSITA could inspect Watson's 1981 paper to search for references to "splitting each triangle through which an intersection line passes," however, this would be fruitless since Watson's

[1] This label is absent from Figure 13 of the '105 patent.

method is for computing a Delaunay triangulation, and not for splitting triangles by intersection lines. Thus, a POSITA would not have been able to answer these questions based on the claims or the specification, and would thus not have been able to ascertain a definite meaning for "modified Watson method."

19. Figures 12A through 12H, cited in NSS' brief, add no clarity.

'961 patent FIGS. 12A-12H	 FIG. 12A Prior Art; FIG. 12B Prior Art; FIG. 12C Prior Art; FIG. 12D Prior Art; FIG. 12E Prior Art; FIG. 12F Prior Art; FIG. 12G; FIG. 12H
'105 patent FIGS. 12A-12H	 FIG. 12A; FIG. 12B; FIG. 12C; FIG. 12D; FIG. 12E; FIG. 12F; FIG. 12G; FIG. 12H

Although the labels below FIGS. 12A-12F in the '961 patent say "Prior Art," unlike the labels below FIGS. 12G-12H, the patent's specification does not describe what, if anything distinguishes the algorithms used to create FIGS. 12A-12F from that used in FIGS. 12G-12H. The '105 patent's figures do not even contain labels to distinguish what is prior art from what is purportedly new. In both patents, the most the specification states of these figures is as follows: "FIGS. 12A through 12H show a Delaunay mesh sequence in which each intersection point is inserted into the mesh step by step. FIG. 13 is the flowchart of Delaunay mesh modified Watson method that created the sequence of FIGS. 12A through 12H." ('961 patent, 3:36-41; '105 patent, 3:23-28.) This text suggests that the *same* technique was used to create FIGS. 12A-12H. A POSITA would not understand from FIGS. 12A-12H or the accompanying text how, if at all, the Watson method had been modified.

20. I understand that NSS argued in its opening claim construction brief that because claim 6 of the '961 patent includes an alleged example of a "modified Watson method," the same term in claim 1 must have a definite meaning. In my opinion, a person of ordinary skill in the art would not have been able to rely on claim 6 to understand claim 1. It is my understanding that, as a dependent claim, claim 6 covers a more specific invention than claim 1, from which it depends. Thus, a POSITA would not have been able to determine what other steps can be involved in a "modified Watson method," or what other algorithms might qualify, for purposes of claim 1.

21. In summary, the term "modified Watson method" is amenable to more than one interpretation, so it is ambiguous and not capable of construction. As the specifications and prosecution histories of the patents fail to inform a POSITA with reasonable certainty as to the scope of the claim, I believe that the term is indefinite.

B. "searching neighboring triangles of the last triangle pair that holds the last intersection point" ('961 and '105 patents)

NSS	Autodesk
[ordinary meaning]	[indefinite] or alternatively, "iteratively searching immediately adjacent triangles of the current intersecting triangle pair to identify a next intersection point"

22. I understand that NSS has argued that the phrase "searching neighboring triangles of the last triangle pair that holds the last intersection point" has a plain and ordinary meaning and requires no construction. I disagree with this position because the phrase "searching neighboring triangles of the last triangle pair that holds the last intersection point" does not and would not have a well-defined meaning to a POSITA. Despite my professional experience, it is unclear to me what this phrase means. It is not one that I have seen used in the field.

23. It is also not clear from the '961 and the '105 patents what this phrase means. This

process is not clearly described in the specifications and prosecution histories. Claim 1 requires:

> ***searching neighboring triangles of the last triangle pair that holds the last intersection point*** to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed.

('961 patent and '105 patent, claim 1 (emphasis added).) This "searching" suggests an iterative process, but it is not clear what it means to iteratively search "neighboring triangles of the last triangle pair." For example, if read literally, "the last triangle pair" might refer to a pair of triangles that were analyzed in a previous iteration. Or, if interpreted logically based on context, it might refer to the current triangle pair under consideration.

24. If the claim language is read literally, as described above, "last triangle pair" refers to a pair of triangles considered in a previous iteration. Then, as an initial matter, after a first pair of triangles is analyzed to identify a first intersection point (and assuming that no other intersection point has yet been identified in another pair of triangles), it is not clear what the "last triangle pair" or "last intersection point" would even mean. These would not be defined.

25. Second, even assuming we can overcome this initialization problem, the phrase "searching neighboring triangles of the last triangle pair that holds the last intersection point" does not explain how to extend an intersection line, as claimed. Let us assume, as NSS does, that each triangle has three neighboring triangles, as shown by the white triangles in Figure 5.

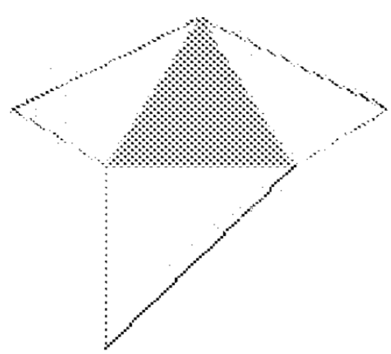
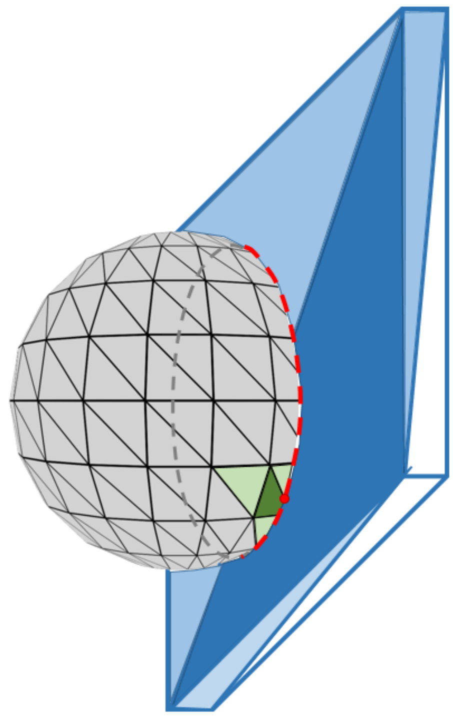


FIG. 5

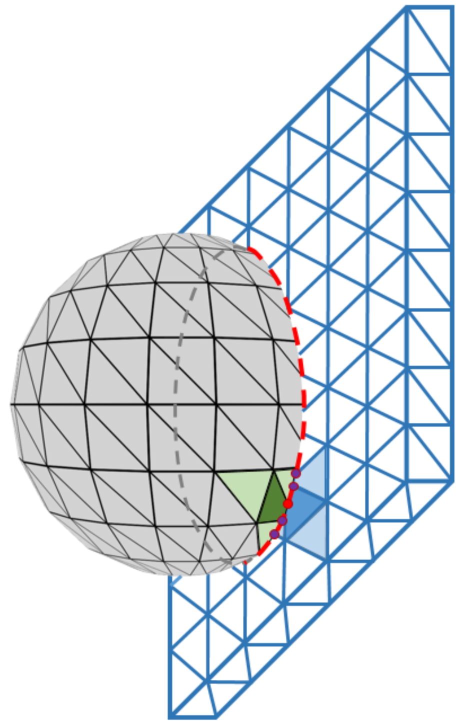
NSS' statement that "a triangle in a three-dimensional space has three neighboring triangles" (ECF No. 35 at 1) is consistent with my understanding that, within the context of the patents, "neighboring" means "immediately adjacent." From the rest of the claim language, however, it is not clear (a) which neighboring triangles, if any, would intersect, and (b) if intersections are found, which should be used to extend the intersection line.

26. Assume that two triangles from two separate objects intersect. In some cases, no additional intersections are found between the neighboring triangles of a first intersecting triangle from a first object and the neighboring triangles of a second intersecting triangle from a second object. The following illustration shows a common example of such a scenario.



In this figure, the dark green triangle on the sphere intersects with the dark blue triangle on the box. Due to the relative sizes and positions of the triangles, there are no intersection points between the dark green triangle's neighboring triangles (shown in light green) and the dark blue triangle's neighboring triangles (shown in light blue). In this scenario, the claim language does not explain how to continue extending the intersection line.

27. Conversely, there may be multiple neighboring triangles with intersection points, as shown below.



Here, the red dot represents an intersection point between dark green and dark blue triangles. The multiple, neighboring triangles of the dark green and dark blue triangles also intersect, yielding additional, potential intersection points (drawn as purple dots). But the claim language, standing alone, does not specify which of those neighboring, intersecting triangles should be used to identify additional intersection points. Nor does the claim specify (where there are multiple potential intersection points for a given pair of neighboring triangles) which of the multiple potential intersection points should be used to extend the intersection line. Thus, the claim language is indefinite.

28. One possible construction of the phrase "searching neighboring triangles of the last triangle pair that holds the last intersection point" would be Autodesk's alternative proposal of "iteratively searching immediately adjacent triangles of the current intersecting triangle pair to identify a next intersection point." Searching neighboring triangles of the current triangle pair under consideration to identify a next intersection point would allow the method to find intersection points that have not already been considered, consistent with Figure 4 and NSS' argument to "search for the first *not-traveled* intersection point." (ECF No. 35 at 4 (emphasis added).) This construction clarifies at least some of the ambiguity in the claim language.

29. In summary, the phrase "searching neighboring triangles of the last triangle pair that holds the last intersection point" fails to inform a POSITA with reasonable certainty as to the scope of the claim. Alternatively, to the extent this phrase can be construed, it should be construed in a manner consistent with Autodesk's proposed construction.

C. "regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets" ('961 and '105 patents)

NSS	Autodesk
construed "new extended triangle sets" to mean "the product of regrouping facets"	[indefinite] or alternatively, "regrouping the extended triangles for a Boolean operation that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets consisting of triangles"

30. I understand that NSS has not sought construction of the phrase "regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets," although NSS has argued that the term "new extended triangle sets" means "the product of regrouping facets." NSS' silence regarding the broader phrase suggests that NSS believes that "regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets" has a plain and ordinary meaning. I disagree. The phrase "regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets" does not have and did not have a well-defined meaning, including at the time of the alleged invention, to a POSITA. Despite my professional experience, it is unclear to me what this phrase means.

31. The individual "steps" in the disputed phrase, such as "copying triangles" or "deleting triangles," were known to a POSITA, but the phrase, as a whole, is ambiguous. First, it is not clear what is meant by "separate steps." For example, it is possible to copy a data structure representing a triangle from a first location in memory to another and then, later, to delete the

triangle from the first location. This would seem to satisfy the claim's notion of "separate steps." On the other hand, one could achieve the same result by moving the data structure representing the triangle from the first memory location to the second in a combined operation. A POSITA would not know whether such a "moving" operation is a single step or consists of "separate steps." Additionally, the term "merging" is not well defined. It is unclear from the specification how "merging" can be separate from the "copying" and "deleting" steps that appear earlier in the claim language. A POSITA would understand that "merging" typically involves copying pieces of information from at least two original locations and then deleting the information from at least one of the original locations. "Separat[ing]" merging from copying and deleting steps might require a different process that is not typically practiced in the art. Thus it is not clear whether the claims cover "merging" as typically understood by a POSITA or some special algorithm for "merging" that the patentee does not explain.

32. The most relevant evidence I could find to explain which Boolean operations the claim intends to cover is not clear. It is found in a passage of the specifications under the heading "Regrouping the Facets." ('961 patent, 8:7-60; '105 patent, 7:36-8:22.) As described in this passage, the purpose of the copying, deleting, reversing, and merging steps is to support "five (5) kinds of Boolean operations." ('961 patent, 8:9; '105 patent, 7:38-39.) These include the Boolean operations of "combination," "intersection," "exclusion," "difference," and "division." Additional ambiguity arises because, as the specifications state, each of these Boolean operations "has a different regrouping procedure." ('961 patent, 8:10; '105 patent, 7:40.) For example, according to the patents, the "combination" operation involves the following steps:

1) Delete obscure triangles of object A.
2) Delete obscure triangles of object B.
3) Merge the triangles of object A and B.

('961 patent, 8:11-18; '105 patent, 7:41-47.) The "copying" and "reversing the normal" steps from the claim language are absent. On the other hand, according to the patents, the "difference" operation involves the following:

1) Delete obscure triangles of object A.
2) Delete NOT obscure triangles of object B.
3) Reverse the normal of every triangle of object B.

4) Merge triangles of object A and B.

('961 patent, 8:39-45; '105 patent, 8:1-8.) Here, the "copying" step is absent. Given that claim 1 recites "copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, *and* merging reserved triangles" (emphasis added), I interpret it to only cover those Boolean operations that include *all* of these steps. Thus, to the extent it is possible to construe the phrase "regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets," the specification supports Autodesk's proposed alternative construction: "regrouping the extended triangles for a Boolean operation that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets consisting of triangles." Of the operations listed in the patent, only the "exclusion" operation seems to involve a copying step, a deleting step, a reversing-the-normal step, and a merging step. But, as explained above, the claim phrase is nevertheless amenable to more than one interpretation. I believe it is indefinite. The specifications and prosecution histories of the patents fail to inform a POSITA with reasonable certainty as to the scope of the claim.

D. "surface trimming" ('961 patent claims 1 and 8)

NSS	Autodesk
map a surface to a BIOpTriangleSet and one of its trimming contours to an extruded shape, checking a triangle as left or right of a trimming contour, deleting left or right side triangles when trimming a surface, plus steps 2, 3, and 6 of the Boolean operation of Figure 4	[ordinary meaning]

33. I understand that NSS has argued that the term "surface trimming" means "map a surface to a BIOpTriangleSet and one of its trimming contours to an extruded shape, checking a triangle as left or right of a trimming contour, deleting left or right side triangles when trimming a surface, plus steps 2, 3, and 6 of the Boolean operation of Figure 4." I disagree with this position because "surface trimming" is a well-known term of art, and I have seen no evidence that the

patentee attempted to redefine "surface trimming" in the asserted patents. Quite simply, "surface trimming" means to trim, or cut a surface along a curve or contour line. For example, one could imagine a hedge with a curved top that is subsequently cut off so that the top of the hedge is parallel to the ground. If this cutting operation were performed on the surface of a three-dimensional model, it would be an example of surface trimming. Claim 8 of the '961 patent is consistent with this understanding. It recites, "checking each triangle whether it is obscure or visible when *trimming a surface patch with a trimming contour*. . . ." ('961 patent, 10:55-57 (emphasis added).)

34. NSS' proposed construction would only confuse a jury. The term "BIOpTriangleSet," incorporated into NSS' construction, is not a known term of art. Indeed, as explained below, despite my professional experience, I have not seen the term BIOpTriangleSet used outside the context of NSS' patents. NSS' proposed construction, which includes not just "BIOpTriangleSet" but also a series of unspecified steps copied from the specification, would be meaningless to a jury.

35. NSS' construction appears to rely on a passage in column 2 of the '961 patent's specification. (*See* '961 patent, 2:43-52 ("The process of the said surface trimming command contains six (6) steps, too. Initially, this system maps a surface to a BIOpTriangleSet . . . .").) This specification passage appears to describe a series of steps that an embodiment of the patent *uses* to carry out surface trimming, but this passage does not purport to redefine "surface trimming." A POSITA would understand that there are other potential algorithms for performing "surface trimming." Thus, NSS' reference to the specification does not change my opinion that "surface trimming" should be given its ordinary meaning in construing the claims of the '961 patent.

E. "regular points" ('961 patent claim 8)

NSS	Autodesk
Regular points are definite, each of which is a Point3dEx located on a surface but not on an intersection line. If we compare "regular points of BiOpTriangleSet" and "points of the intersection lines," then its meaning is clear.	[indefinite]

36. I understand that NSS has argued that "regular points" are "definite, each of which is a Point3dEx located on a surface but not on an intersection line." I disagree with this position. I first note that "regular points" does not have and did not have a single, well-defined meaning, including at the time of the alleged invention, to a POSITA. For example, depending on context, "regular points" might refer to points that are evenly spaced in a grid. Alternatively, depending on context of usage, "regular points" might refer to the result of using a "regular expression" or to points that are not "special," *i.e.*, points that are "regular."

37. Since "regular points" does not have a customary meaning to a POSITA, I searched for its meaning in the '961 patent itself. When the term "regular points" is read in the context of the claims and specification, a POSITA would not understand what is meant by "regular points," as claimed. The term "regular points" appears only twice in the patent, in claims 8 and 18, as part of the phrase "setting m_ID of regular points of BIOpTriangleSet of the concerned patch to be 0." This language indicates only that "regular points" must be "of BIOpTriangleSet" and must have an associated "m_ID." The claims do not describe what "regular points" are or provide any meaningful structural limitations. For example, the claim language could be interpreted to require taking a set of points and then setting their "m_ID" to be zero, thus creating "regular points," or it could be interpreted to require taking a set of "regular points" and setting their "m_ID" to be zero. Moreover, the term "regular points" is not even mentioned, let alone defined, anywhere in the specification. Based on the specification, it is unclear whether there are any points that are not "regular points." The specification does not provide objective boundaries for a POSITA to determine whether a point is a regular point or not. Thus, the term is ambiguous and not capable of construction. As the specification and

prosecution history of the '961 patent fail to inform a POSITA with reasonable certainty as to the scope of the claim, I believe that the term is indefinite.

F. BIOpTriangleSet" ('961 patent)

NSS	Autodesk
A BIOpTriangleSet is a triangle set built for Boolean operations or surface trimmings.	[indefinite] or alternatively, "a set of triangles represented by a data structure including a set of three-dimensional points and a set of triangle data structures, wherein each triangle data structure of the set of triangle data structures comprises information related to points, a plane, a normal, three immediately adjacent triangles and an index of the respective triangle"

38. I understand that in its Patent Local Rule 4-2 disclosures, NSS previously argued that the term "BIOpTriangleSet" should be given its "ordinary meaning," but that in the Joint Claim Construction and Prehearing Statement, NSS has argued that the term "BIOpTriangleSet" means "a triangle set built for Boolean operations or surface trimmings." To the extent that NSS believes its current proposal reflects the "ordinary meaning" of "BIOpTriangleSet," I disagree with NSS because "BIOpTriangleSet" is not a known term of art. In my professional experience, this is the first time I have seen this term used. It does not have and did not have an ordinary and customary meaning, including at the time of the alleged invention, to a POSITA.

39. Since "BIOpTriangleSet" does not have a meaning to a POSITA, I searched for its meaning in the '961 patent itself. The closest thing I could find to a definition of a "BIOpTriangleSet" was not actually written in English, but instead, in pseudo-code (i.e., text that resembles computer source code but would not actually run on a computer):

> The Boolean Operation method described in this invention defined three (3) key classes: BIOpTriangleSet, Triangle3dEx, and Point3dEx.

```
class BIOpTriangleSet
{
    DataSet<Point3dEx>          m_PointSet;
    DataSet<Triangle3dEx>       m_TriangleSet;
};
class Point3dEx : Point3d
{
        DataTypeII          m_ID; // position and sequence index
        DataTypeIII         m_X, m_Y, m_Z; // DataType III may
                            be different from DataTypeI
};
class Triangle3dEx : Triangle3d
{
        Point3dEx           *m_Points[3];
        DataTypeII           m_ID;
        Plane                m_Plane;
        DataTypeIV           m_Normal[3];
        Triangle3dEx        *m_NeigTri[3]; // neighboring triangles
};
```

('961 patent, 4:37-57.) The patent states: "The Boolean Operation method described in this invention *defined* three (3) key classes: BIOpTriangleSet, Triangle3dEx, and Point3dEx." (*Id*. at 4:37-39 (emphasis added).) The '961 patent does not attempt to "define" a BIOpTriangleSet in plain English. Instead, from the pseudo-code above, a POSITA could gather that each BIOpTriangleSet is defined to include a "Point3dEx" and a "Triangle3dEx." These terms, just like "BIOpTriangleSet" are not terms of art; they are meaningless outside the context of the asserted patents. From context, I interpret the above pseudo-code to mean that a "Point3dEx" is a set of three-dimensional points and that a "Triangle3dEx" is a set of triangle data structures. Furthermore, the above pseudo-code indicates that each "Triangle3dEx" includes information related to a set of points, an index (m_ID), a plane, a normal, and three "neighboring triangles." Based on the pseudo-code above, the most reasonable construction of "BIOpTriangleSet" is "a set of triangles represented by a data structure including a set of three-dimensional points and a set of triangle data structures, wherein each triangle data structure of the set of triangle data structures comprises information related to points, a plane, a normal, three immediately adjacent triangles and an index of the respective triangle."

40. But I emphasize that this construction is based on pseudo-code and that the

patentee did not attempt to define "BIOpTriangleSet" in plain English. Thus, it is not clear from the specification whether "BIOpTriangleSet" refers to the precise data structure "defined" in the pseudo-code or to something broader. The specification and prosecution history of the patent fail to inform a POSITA with reasonable certainty as to the scope of the claim.

G. "according to m_ID of the member m_Points of each triangle, deciding whether it is a boundary triangle" ('961 patent)

NSS	Autodesk
Definite. A boundary triangle is a triangle, one or more points (vertices) are located on an intersection line, their ID's are not zero.	[indefinite]

41. I understand that in its Patent Local Rule 4-2 disclosures, NSS previously argued that the phrase "according to m_ID of the member m_Points of each triangle, deciding whether it is a boundary triangle" should be given its "ordinary meaning." I understand that in the Joint Claim Construction and Prehearing Statement, NSS now argues that "[a] boundary triangle is a triangle, one or more points (vertices) are located on an intersection line, their ID's are not zero." To the extent that NSS believes its current proposal reflects the "ordinary meaning" of the disputed phrase, I disagree with this position because "according to m_ID of the member m_Points of each triangle, deciding whether it is a boundary triangle" is not a known term of art. Additionally, the term "boundary triangle," as used in the claims and specification, does not convey a clear and ordinary meaning to a POSITA. Despite my professional experience, it is unclear to me what the broader, disputed phrase means. It is not one that I have seen used. It does not have and did not have an ordinary and customary meaning, including at the time of the alleged invention, to a person of ordinary skill in the art ("POSITA").

42. Since "according to m_ID of the member m_Points of each triangle, deciding whether it is a boundary triangle" does not have meaning to a POSITA, and this phrase could have more than one interpretation, I searched for its meaning in the '961 patent itself. While the disputed phrase is mentioned in claims 8 and 18, as well as in a single passage in the specification, it is not defined anywhere. The disputed phrase is mentioned as part of the following passage:

> "When performing surface trimming, this system calls the followings procedure to determine whether a triangle is obscure.
> 1) Set the member m_ID of each Point3dEx of BIOpTriangleSet of the concerned surface patch to be 0.
> 2) Mark m_ID of Point3dEx of the intersection lines of the said patch in ascending or descending order, which is depending on whether the said line and trimming contour are in the same direction, for example, both of them are counterclockwise.
> 3) *According to m_ID of the member m_Points of each triangle, decide whether it is a boundary triangle*.
> 4) For each boundary triangle, decide it is to the left or right side of the trimming contour, and set its neighbors that are not boundary ones to be left or right."

('961 patent, 7:58-8:5 (emphasis added).) The specification is nearly identical to the claim language and does not shed any light on what it actually means. For example, it is not clear from the claims or specification which values of m_ID will result in a triangle being declared a "boundary triangle." Additionally, neither the claim language nor the specification makes clear which "m_Points" need to be analyzed—all of them, or some undefined subset. A POSITA would not know what type of analysis of "m_ID's" and "m_Points" fits within the scope of the claim language.

43. The specification says to decide whether a triangle is a boundary triangle without providing any guidance as to what a boundary triangle is or how to determine whether a triangle is a boundary triangle. As the specifications and prosecution histories of the patents fail to inform a POSITA with reasonable certainty as to the scope of the claim, I believe that the phrase "according to m_ID of the member m_Points of each triangle, deciding whether it is a boundary triangle" is indefinite.

I declare under penalty of perjury under the laws of the United States that the foregoing is true and correct. Executed at West Lafayette, Indiana, on May 11, 2020.

Daniel G. Aliaga

Claims 2 and 12 recites the limitation, “the resultants” in line 3, are considered indefinite because the term “the resultants” lacks clear antecedent basis in the claim.

Claims 3 and 13 recite the limitation, “the resultants” in line 3, are considered indefinite because the term “the resultants” lacks clear antecedent basis in the claim.

Claims 4 and 14 recite the limitation, “wherein building intersection lines” in line 1, however, this limitation is considered indefinite because it lacks antecedent basis in the claim. That is, the wherein clause appears to be further defining “building intersection lines”, however, there is no positively recited limitation that is previously directed to building intersection lines.

Claim 4 recites the limitation, “carries out edge-triangle intersection calculations so that intersection points are exact and the intersection lines are not approximating curved edge lines” in lines 2-3, however, this limitation renders the claim indefinite because the intended result is being claimed without setting forth the method steps required to arrive at said result. Therefore, the examiner cannot ascertain the boundaries of the claimed invention.

Claims 5 and 15 recite the limitation, “wherein searching an intersection point” in line 1, however, this limitation is considered indefinite because it lacks antecedent basis in the claim. That is, the wherein clause appears to be further defining “searching an intersection point”, however, there is no positively recited limitation that is previously directed to searching an intersection point”.

Claim 6 recites the limitation, “modified Watson method” in line 3, however, the claim is considered indefinite because the meets and bounds of “modified Watson method” are not clearly set forth. That is, the claim fails to positively recite the steps that modify the Watson method, therefore, the examiner cannot ascertain the scope of the modified Watson method.

Claims 7 and 17 recites the limitation, “the step checking whether…”, however this limitation is considered indefinite because it lacks antecedent basis in the claim.

Claims 8 and 18 recites the limitation, “the step checking whether…”, however this limitation is considered indefinite because it lacks antecedent basis in the claim.

EXAMINER'S AMENDMENT

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in an interview with Shangwen Cao on 08/28/2018.

The application has been amended as follows:

IN THE CLAIMS: Please amend claims 1 and 11 as follows:

1. (currently amended) A method that performs immediate Boolean operations using geometric facets of geometric objects implemented in a computer system and operating with a computer, the method comprising:

mapping rendering facets to extended triangles that contain neighbors;

building intersection lines starting with and ending with searching for the first pair of triangles that hold a start point of an intersection line by detecting whether two minimum bounding boxes overlap and performing edge-triangle intersection calculations for locating an intersection point, then searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed;

splitting each triangle through which an intersection line passes using modified Watson method, wherein the modified Watson method includes removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion, and a middle portion;

checking each triangle whether it is obscure or visible for Boolean operations or for surface trimming;

regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets; and

mapping extended triangles to rendering facets.

11. (currently amended) A computer system consisting of hardware and software that performs immediate Boolean operations using rendering facets of geometric objects, the system comprising:

a computer with input devices for entering data and commands, and a display device showing user interface, geometric objects, and additional data, having a medium storing geometric data and instructions that make up of a software system, or having a microchip or integrated circuit embedding partially or totally the instructions, and a processor that executes the steps of:

creating, modifying or loading primary geometric objects including swept and extruded ones and relocating them at different positions or orientations with input devices of the computer;

selecting two of the geometric objects;

mapping rendering facets to extended triangles that contain neighbors;

building intersection lines starting with and ending with searching for the first pair of triangles that hold a start point of an intersection line by detecting whether two minimum bounding boxes overlap and by performing edge-triangle intersection calculations for locating an

Claim Rejections - 35 USC § 112

The following is a quotation of 35 U.S.C. 112(b):

(b) CONCLUSION.—The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention.

The following is a quotation of 35 U.S.C. 112 (pre-AIA), second paragraph:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1-20 are rejected under 35 U.S.C. 112(b) or 35 U.S.C. 112 (pre-AIA), second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the inventor or a joint inventor, or for pre-AIA the applicant regards as the invention.

Claim 1 recites the limitation, “performing edge triangle intersection”, however, the claim is considered indefinite because the examiner cannot ascertain the boundaries of the claimed limitation. That is, no specific steps are recites that perform the function of “edge triangle intersection”.

Claim 1 recites the limitation, “extending the intersection lines until they get closed by searching neighboring triangles” in lines 5-6. This limitation renders the claim indefinite because the antecedent basis for “the intersection lines” is not clear. A previous limitation discloses “an intersection line”, which is a singular line, whereas the limitation in question is a plurality of lines. Furthermore, the limitation of “until they get closed” is not clear. What is causing the closure of the intersection lines? The nexus between “extending the intersection lines” and “searching neighboring triangles” is also not clearly set forth. The examiner is not able to ascertain the scope of the claimed invention.

Claim 1 recites the limitation, “deleting, and reserving triangles” in line 7, however, this limitation in considered indefinite because the scope of the limitation is not clear. That is, is the limitation requiring a single step of deleting and reserving or separate steps of deleting triangles and reserving triangles?

the objects.

For claims 10 and 20 the limitation, “the Boolean operation resultant”, the applicant amended the limitation with the term “the extended triangles” that has antecedent basis in claims 1 and 11.

Claims with Remarks

1. A method that performs immediate Boolean operations using geometric facets of geometric objects implemented in a computer system and operating with a computer, the method comprising: mapping rendering facets to extended triangles that contain neighbors; building intersection lines starting with and ending with searching for the first pair of triangles that hold a start point of an intersection line by detecting whether two minimum bounding boxes overlap and performing edge-triangle intersection calculations for locating an intersection point, ~~extending the intersection lines until they get closed by searching neighboring triangles or all triangles are traversed;~~ then searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed; splitting triangles through which ~~intersection lines pass~~ an intersection line passes using modified Watson method; checking each triangle whether it is obscure or visible for Boolean operations or for surface trimming; regrouping facets~~, deleting, and reserving~~ in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles according to Boolean operation types; and mapping extended triangles to rendering facets.

2. The method of claim 1 wherein any Boolean operations, ~~including combination, division, intersection, difference, and exclusion, use rendering facets of the geometric objects to create new geometric objects and the extended triangles are immediately~~

Claim Rejections - 35 USC § 112

The following is a quotation of 35 U.S.C. 112(b):

(b) CONCLUSION.—The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the inventor or a joint inventor regards as the invention.

The following is a quotation of 35 U.S.C. 112 (pre-AIA), second paragraph:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 1-20 are rejected under 35 U.S.C. 112(b) or 35 U.S.C. 112 (pre-AIA), second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the inventor or a joint inventor, or for pre-AIA the applicant regards as the invention.

Claim 1 recites the limitation, “performing edge triangle intersection”, however, the claim is considered indefinite because the examiner cannot ascertain the boundaries of the claimed limitation. That is, no specific steps are recites that perform the function of “edge triangle intersection”.

Claim 1 recites the limitation, “extending the intersection lines until they get closed by searching neighboring triangles” in lines 5-6. This limitation renders the claim indefinite because the antecedent basis for “the intersection lines” is not clear. A previous limitation discloses “an intersection line”, which is a singular line, whereas the limitation in question is a plurality of lines. Furthermore, the limitation of “until they get closed” is not clear. What is causing the closure of the intersection lines? The nexus between “extending the intersection lines” and “searching neighboring triangles” is also not clearly set forth. The examiner is not able to ascertain the scope of the claimed invention.

Claim 1 recites the limitation, “deleting, and reserving triangles” in line 7, however, this limitation in considered indefinite because the scope of the limitation is not clear. That is, is the limitation requiring a single step of deleting and reserving or separate steps of deleting triangles and reserving triangles?

the objects.

For claims 10 and 20 the limitation, "the Boolean operation resultant", the applicant amended the limitation with the term "the extended triangles" that has antecedent basis in claims 1 and 11.

Claims with Remarks

1. A method that performs immediate Boolean operations using geometric facets of geometric objects implemented in a computer system and operating with a computer, the method comprising: mapping rendering facets to extended triangles that contain neighbors; building intersection lines starting with and ending with searching for the first pair of triangles that hold a start point of an intersection line by detecting whether two minimum bounding boxes overlap and performing edge-triangle intersection calculations for locating an intersection point, ~~extending the intersection lines until they get closed by searching neighboring triangles or all triangles are traversed;~~ then searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed; splitting triangles through which ~~intersection lines pass~~ an intersection line passes using modified Watson method; checking each triangle whether it is obscure or visible for Boolean operations or for surface trimming; regrouping facets~~, deleting, and reserving~~ in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles according to Boolean operation types; and mapping extended triangles to rendering facets.

2. The method of claim 1 wherein any Boolean operations, ~~including combination, division, intersection, difference, and exclusion, use rendering facets of the geometric objects to create new geometric objects and the extended triangles are immediately~~

to geometric object *A*; building a line *l: p=c+t*N* passing through the centroid *c* and along the normal of T_a; for each triangle T_b of object *B,* checking whether *l* intersects with T_b at an interior point and adding *t* to a depth buffer, *t-Buffer*; and setting T_a to be ~~"obscure"~~ obscure when ~~the size of negative *t* equals to that of positive *t* in *t-Buffer*~~ the result of the size of negative *t* in *t-Buffer* modulo 2 equals to the result of the size of positive *t* in *t-Buffer* modulo 2 and the result is not zero.

8. The method of claim 1 wherein ~~the step checking whether a triangle is "obscure"~~ checking each triangle whether it is obscure or visible when trimming a surface patch with a trimming contour further composing: setting *m_ID* of regular points of *BlOpTriangleSet* of the concerned patch to be 0 and *m_ID* of points of the intersection lines of the said patch in ascending or descending order, which is depending on whether the said line and the trimming contour are in the same direction; according to *m_ID* of the member *m_Points* of each triangle, deciding whether it is a boundary triangle; for each boundary triangle, determining it is to the left or right side of the trimming contour, and setting its neighbors that are not boundary ones to be ~~"left" or "right"~~ left or right.

9. The method of claim 1 wherein a Boolean operation~~, such as~~ that is a combination, an intersection, an exclusion, a difference, or a division, ~~reserves visible, or obscure facets, or a mixture of visible and obscure facets for constructing its operational result~~ regroups facets for constructing its operational result using one or more steps of: deleting obscure or visible triangles of an object, copying obvious triangles of an object to a buffer or copying triangles from a buffer to an object, reversing the normal of each triangle of an object, and merging the triangles of the objects to form new extended triangle sets.

10. The method of claim 1 wherein the ~~Boolean operation result is~~ extended triangles are directly mapped to rendering ~~data~~ facets for being displayed and providing data to next Boolean operations.

		2018; Response to Non-Final Office Action, dated March 4, 2018; Claim Amendment and Remarks, dated April 15, 2018. For ’105: *See also* Non-Final Office Action, dated February 8, 2018; Response to Non-Final Office Action, dated March 4, 2018; Claim Amendment and Remarks, dated May 3, 2018. EXTRINSIC EVIDENCE: Testimony from expert Dr. Daniel Aliaga that to a person of ordinary skill in the art, “searching neighboring triangles of the last triangle pair that holds the last intersection point” does not and did not have an ordinary and customary meaning, including at the time of the alleged invention, and is indefinite.
mapping rendering facets to extended triangles that contain neighbors Found in: ’961 Patent, Claim 1 ’105 Patent, Claim 1	PROPOSED CONSTRUCTION: adding neighboring triangles to each triangle in TriangleSet to form the set BIOpTriangleSet. INTRINSIC EVIDENCE ‘961, 1:66-2:11; 5:7-40; 8:64-9:2. ‘105, 1:63-2:7; 4:56-5:19; 8:26-31	PROPOSED CONSTRUCTION: <u>triangulating rendering facets to decompose them into triangles that have immediately adjacent triangles</u> INTRINSIC EVIDENCE: “This invention provides a set of data structures and algorithms for performing Boolean operations, which are used to build complex geometric models and work directly on triangles decomposed from geometric facets used as rendering data by computer hardware and rendering functions like OpenGL libraries. A geometric shape, for example, a sphere, a cone, a cylinder, a box, triangular facets, an extruded or swept object, and a surface patch, is triangulated to build a set, noted as

AUTODESK®

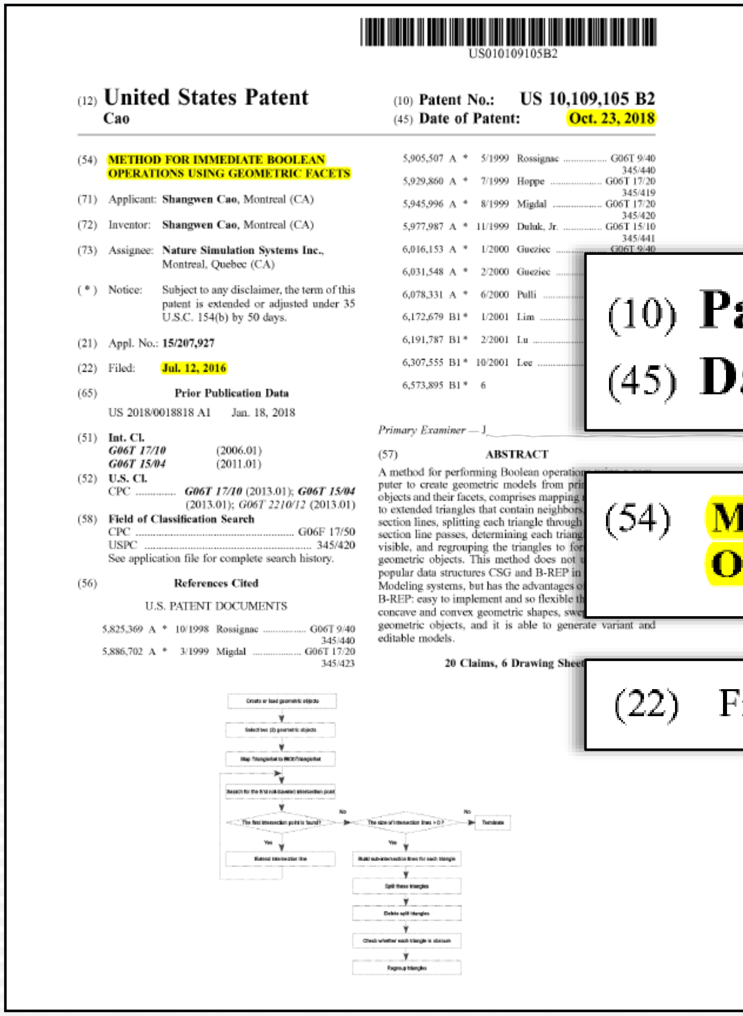
Nature Simulation Systems, Inc. v. Autodesk, Inc.

Case No. 3:19-cv-03192-SK

Technical Tutorial

May 26, 2020

U.S. Patent No. 10,109,105



US010109105B2

(12) United States Patent
Cao

(10) Patent No.: US 10,109,105 B2
(45) Date of Patent: Oct. 23, 2018

(54) METHOD FOR IMMEDIATE BOOLEAN OPERATIONS USING GEOMETRIC FACETS

(71) Applicant: Shangwen Cao, Montreal (CA)

(72) Inventor: Shangwen Cao, Montreal (CA)

(73) Assignee: Nature Simulation Systems Inc., Montreal, Quebec (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 50 days.

(21) Appl. No.: 15/207,927

(22) Filed: Jul. 12, 2016

(65) Prior Publication Data

US 2018/0018818 A1 Jan. 18, 2018

(51) Int. Cl.
G06T 17/10 (2006.01)
G06T 15/04 (2011.01)

(52) U.S. Cl.
CPC G06T 17/10 (2013.01); G06T 15/04 (2013.01); G06T 2210/12 (2013.01)

(58) Field of Classification Search
CPC G06F 17/50
USPC 345/420
See application file for complete search history.

(56) References Cited

U.S. PATENT DOCUMENTS

5,825,369 A * 10/1998 Rossignac G06T 9/40 345/440
5,886,702 A * 3/1999 Migdal G06T 17/20 345/423
5,905,507 A * 5/1999 Rossignac G06T 9/40 345/440
5,929,860 A * 7/1999 Hoppe G06T 17/20 345/419
5,945,996 A * 8/1999 Migdal G06T 17/20 345/420
5,977,987 A * 11/1999 Duluk, Jr. G06T 15/10 345/441
6,016,153 A * 1/2000 Gueziec G06T 9/40
6,031,548 A * 2/2000 Gueziec
6,078,331 A * 6/2000 Pulli
6,172,679 B1 * 1/2001 Lim
6,191,787 B1 * 2/2001 Lu
6,307,555 B1 * 10/2001 Lee
6,573,895 B1 * 6

Primary Examiner — J

(57) ABSTRACT

A method for performing Boolean operation ... puter to create geometric models from pri... objects and their facets, comprises mapping ... to extended triangles that contain neighbors ... section lines, splitting each triangle through ... section line passes, determining each triang... visible, and regrouping the triangles to for... geometric objects. This method does not ... popular data structures CSG and B-REP in ... Modeling systems, but has the advantages o... B-REP: easy to implement and so flexible th... concave and convex geometric shapes, swe... geometric objects, and it is able to generate variant and editable models.

20 Claims, 6 Drawing Sheets

(10) Patent No.: US 10,109,105 B2
(45) Date of Patent: Oct. 23, 2018

(54) METHOD FOR IMMEDIATE BOOLEAN OPERATIONS USING GEOMETRIC FACETS

(22) Filed: Jul. 12, 2016

Asserted Claim: 1

U.S. Patent No. 10,120,961

US010120961B2

(12) United States Patent
Cao

(10) Patent No.: US 10,120,961 B2
(45) Date of Patent: *Nov. 6, 2018

(54) METHOD FOR IMMEDIATE BOOLEAN OPERATIONS USING GEOMETRIC FACETS

(71) Applicant: Shangwen Cao, Montreal (CA)

(72) Inventor: Shangwen Cao, Montreal (CA)

(73) Assignee: Nature Simulation Systems Inc., Montreal, Quebec (CA)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.
This patent is subject to a terminal disclaimer.

(21) Appl. No.: 15/840,052

(22) Filed: Dec. 13, 2017

(65) Prior Publication Data
US 2018/0113958 A1 Apr. 26, 2018

Related U.S. Application Data

(63) Continuation-in-part of application No. 15/207,927, filed on Jul. 12, 2016.

(51) Int. Cl.
G06F 17/50 (2006.01)
G06T 17/10 (2006.01)
G06F 17/10 (2006.01)
G06T 17/20 (2006.01)
G06F 9/30 (2018.01)

(52) U.S. Cl.
CPC *G06F 17/50* (2013.01); *G06F 9/30029* (2013.01); *G06F 17/10* (2013.01); *G06T 17/10* (2013.01); *G06T 17/20* (2013.01); *G06T 2210/21* (2013.01)

(58) Field of Classification Search
CPC G06F 17/50
USPC 345/420
See application file for complete search history.

(56) References Cited

U.S. PATENT DOCUMENTS

5,825,369 A * 10/1998 Rossignac G06T 9/40
5,886,702 A * 3/1999 Migdal
5,905,507 A * 5/1999 Rossignac
5,929,860 A * 7/1999 Hoppe
(Continued)

OTHER PUBLICATION

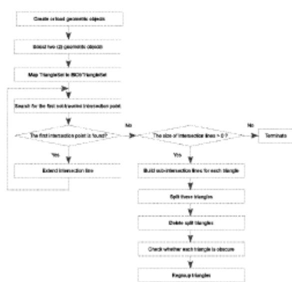
Landier, "Boolean operat... International Meshing R...

Primary Examiner — Jason C Olson

(57) ABSTRACT

A method for performing Boolean opera... puter to create geometric models from ... objects and their facets, comprises mappi... to extended triangles that contain neighb... section lines, splitting each triangle throu... section line passes, determining each ... obscure, and regrouping the facets to ... geometric objects. This method does n... popular data structures CSG and B-REP... Modeling systems, but has the advantage... B-REP: easy to implement and flexible... united method for solid modeling and surface modeling systems, and it is able to generate variant and editable models.

20 Claims, 6 Drawing Sh...



(10) Patent No.: US 10,120,961 B2
(45) Date of Patent: *Nov. 6, 2018

(54) METHOD FOR IMMEDIATE BOOLEAN OPERATIONS USING GEOMETRIC FACETS

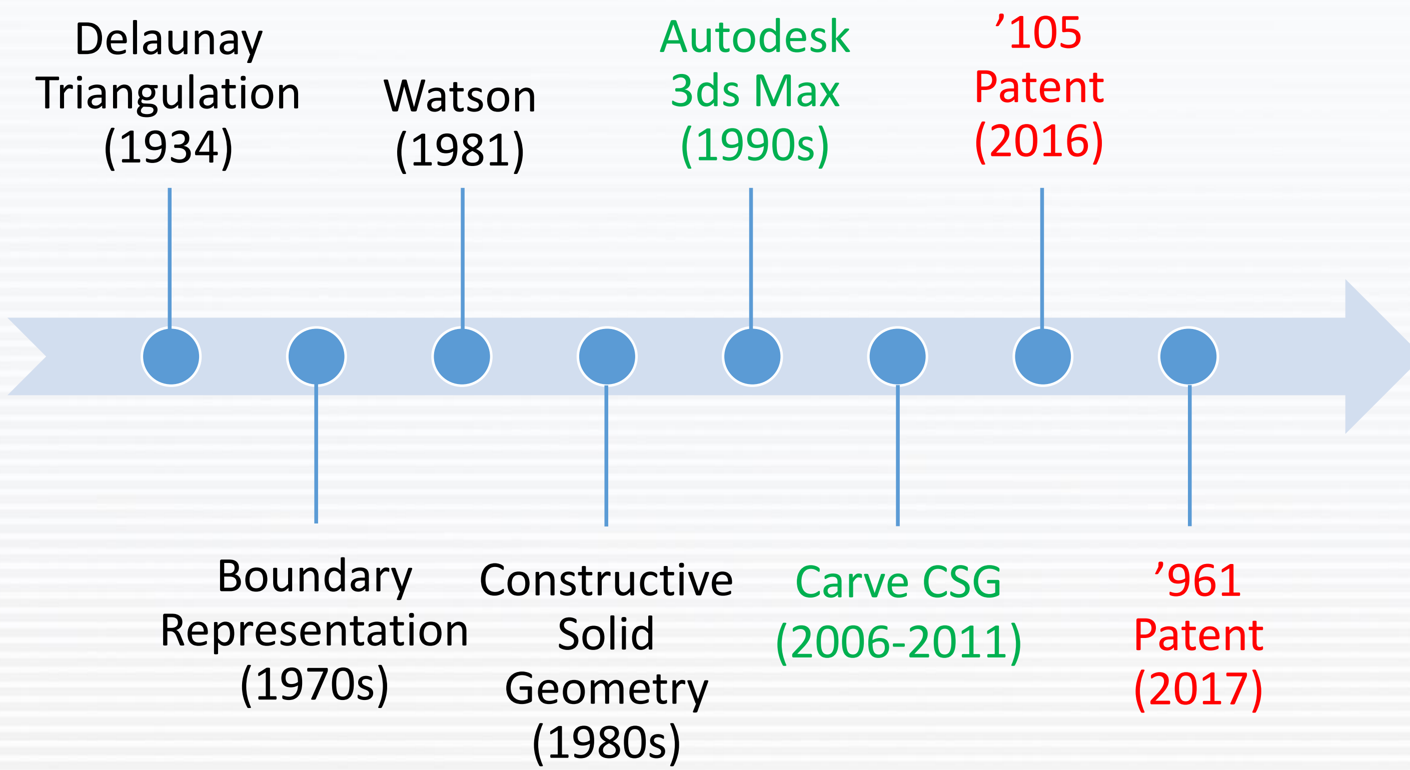
(22) Filed: Dec. 13, 2017

Asserted Claims: 1, 8

Contents

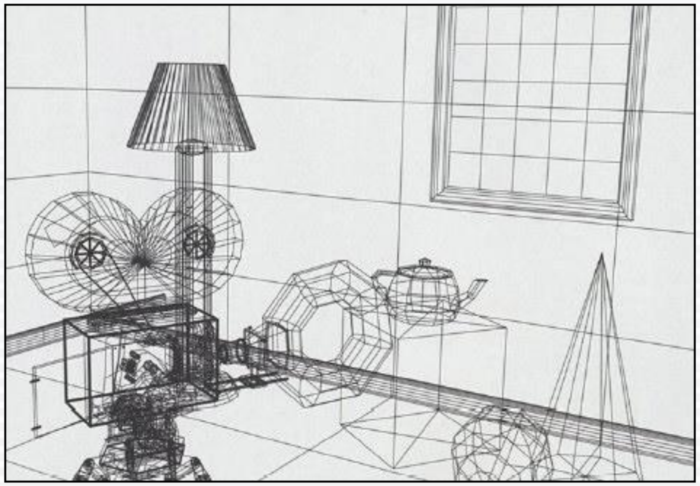
Contents

History of 3-D Modeling



Introduction to 3-D Modeling

Building and displaying 3-dimensional objects on a 2-dimensional screen



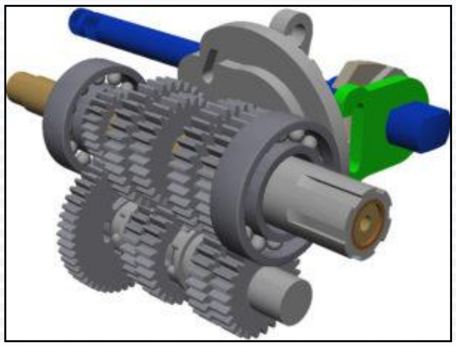
1. Create 3-D design of scene

2. Use facets to add shading & lighting

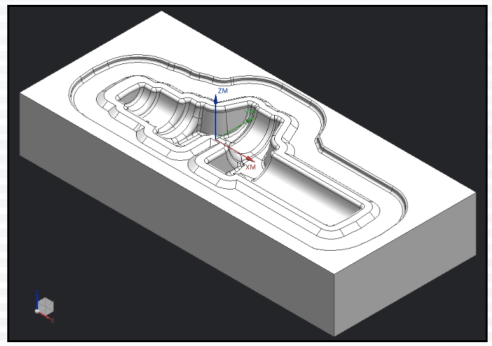
Solid Modeling

In solid modeling:

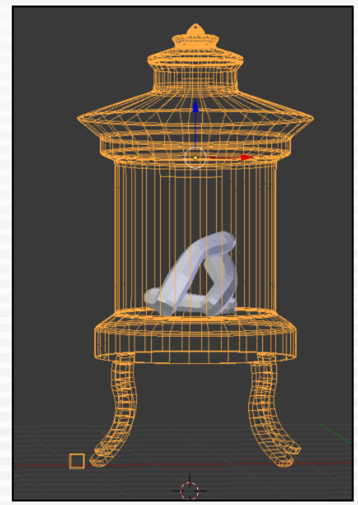
- 3-D objects are designed using many different tools
- 3-D objects can be represented using numerous formats



https://samsonyslcad.com/3d-solid-modeling/

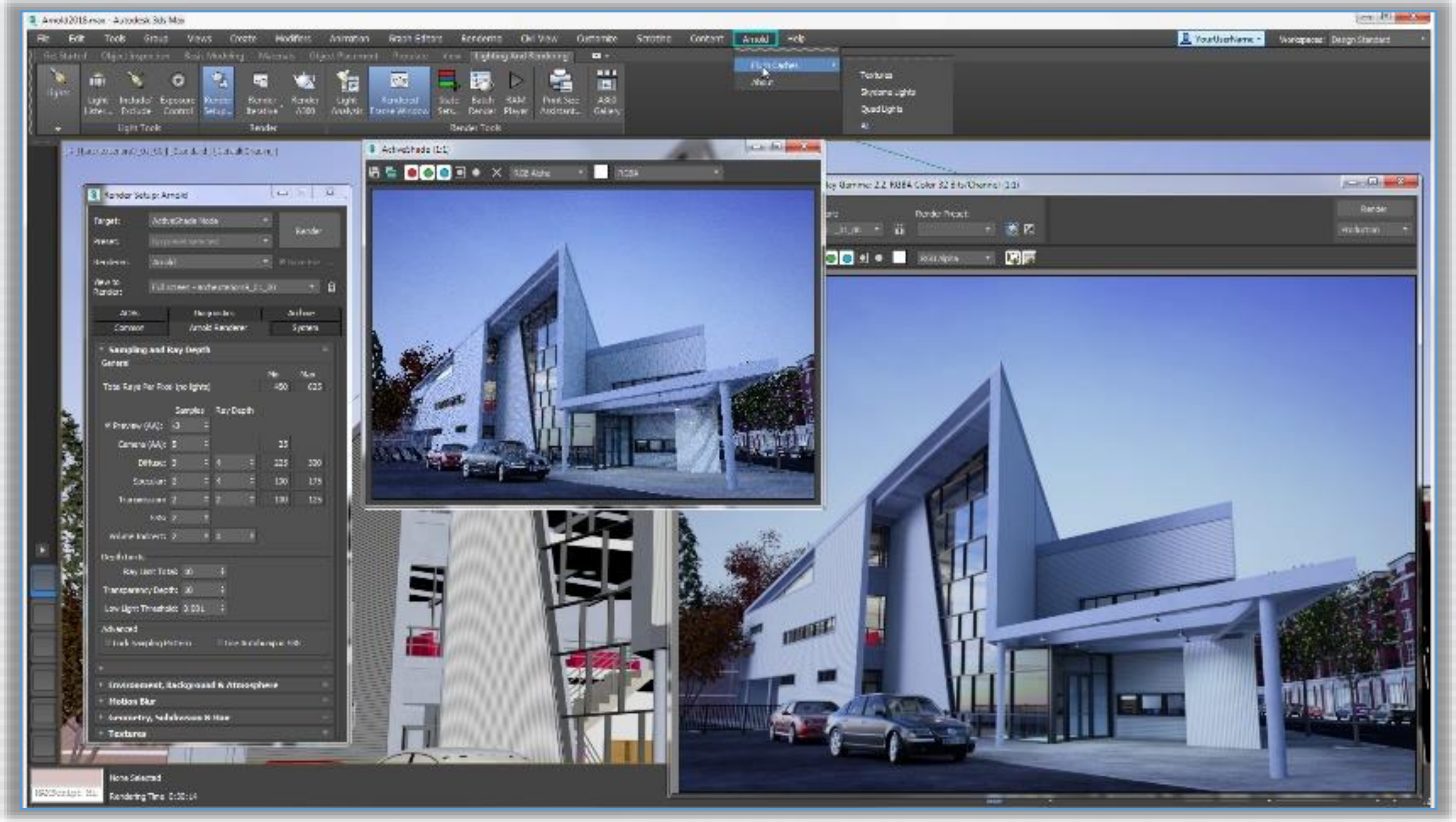


https://www.fiverr.com/chiraghiwill/engineering-drawings-and-solid-modeling



https://blender.stackexchange.com/questions/12007/cant-get-out-of-wireframe-mode

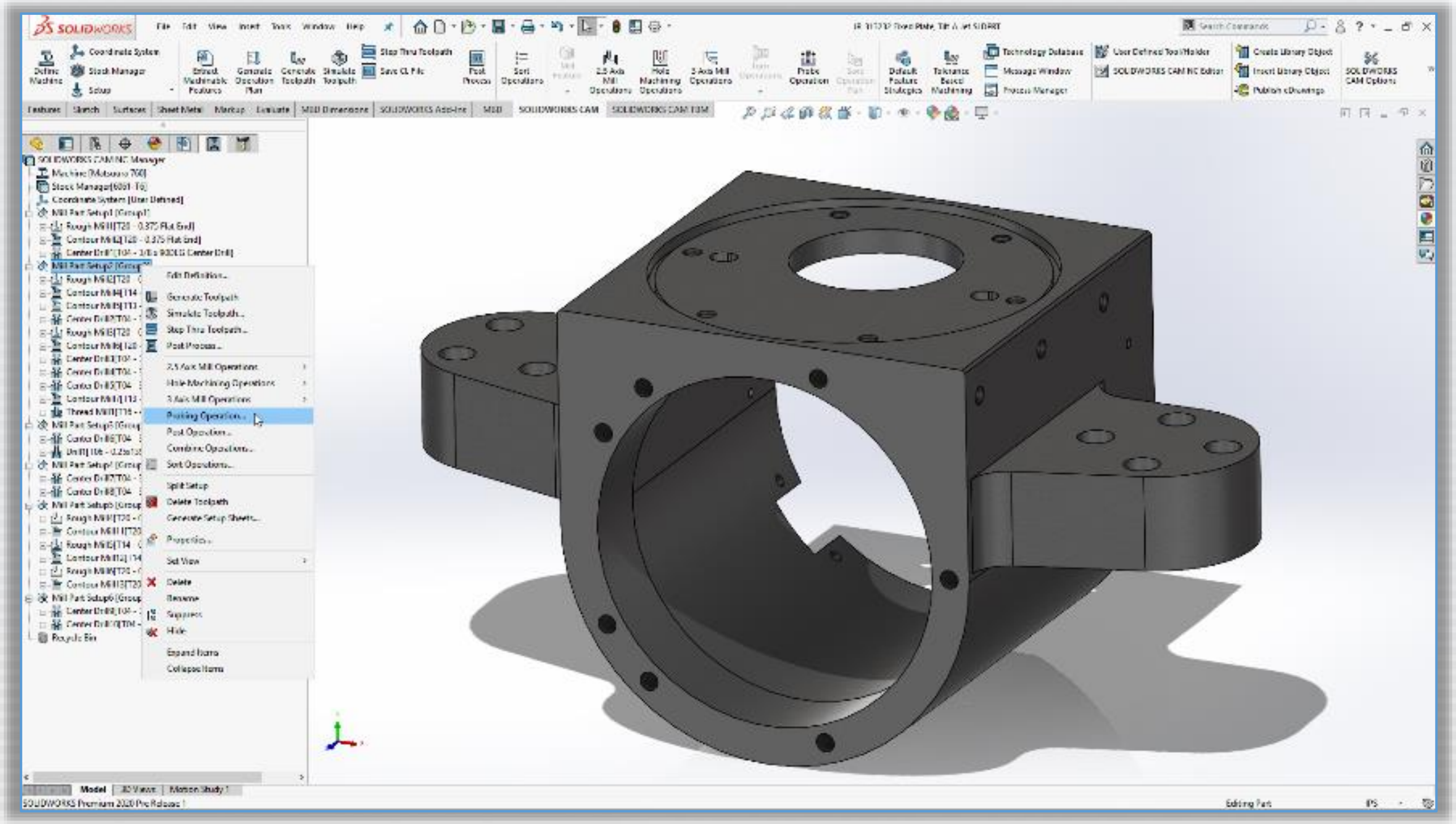
Examples of 3-D Modeling Products



Autodesk 3ds Max

https://area.autodesk.com/blogs/the-3ds-max-blog/3dsmax2018/

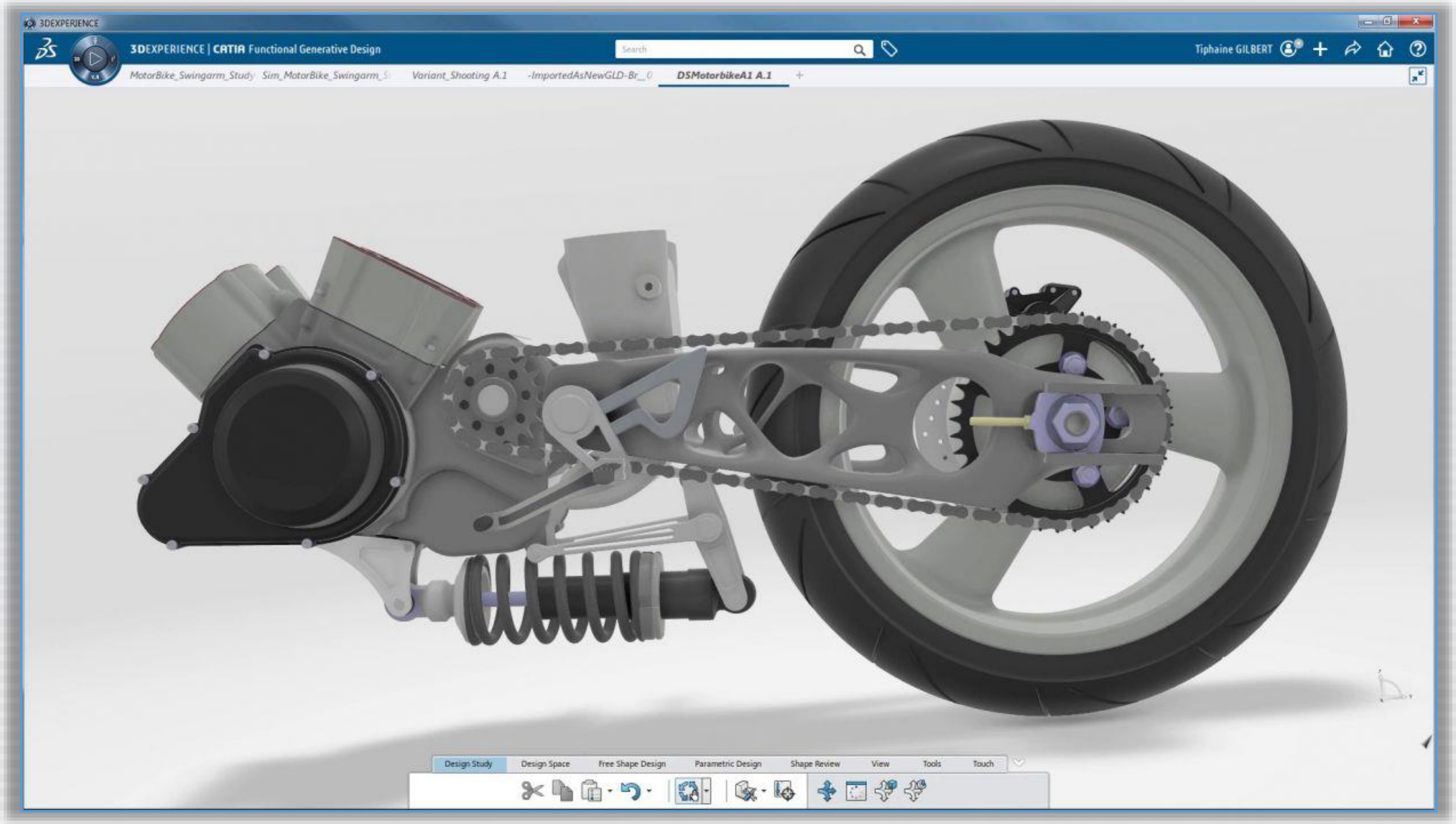
Examples of 3-D Modeling Products



SolidWorks CAM

https://blogs.solidworks.com/solidworksblog/2019/10/whats-new-in-solidworks-cam-2020.html

Examples of 3-D Modeling Products



Dassault CATIA

https://discover.3ds.com/optimize-your-catia-v5-parts-touch-button

Contents

Key Technical Concepts

(57) **ABSTRACT**

A method for performing Boolean operations using a computer to create geometric models from primary geometric objects and their facets, comprises mapping rendering facets to extended triangles that contain neighbors; building intersection lines, splitting each triangle through which an intersection line passes, determining each facet is visible or obscure, and regrouping the facets to form one or more geometric objects. This method does not utilize the most popular data structures CSG and B-REP in CAD/CG/Solid Modeling systems, but has the advantages of both CSG and B-REP: easy to implement and flexible. Additionally it is a united method for solid modeling and surface modeling systems, and it is able to generate variant and editable models.

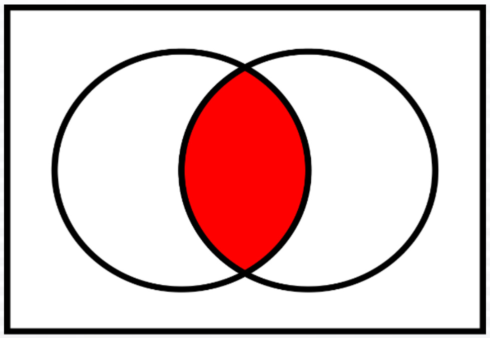
('961 Patent at Abstract.)

Contents

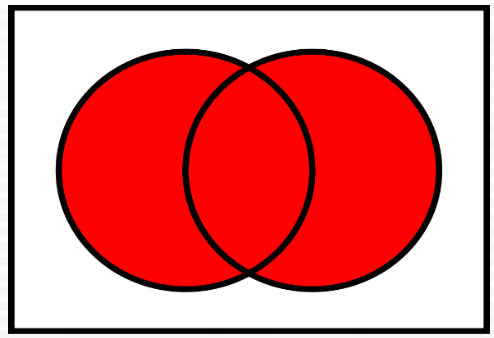
- 3-D Modeling Overview
- Technical Concepts Used in 3-D Modeling
  - Boolean Operations
  - Constructive Solid Geometry
  - Boundary Representation
  - Facets
  - Triangulation
- Other Concepts in Patents
- Autodesk 3ds Max

Boolean Operations

- Boolean operations use logic on two objects to produce a third object
- Venn diagrams can represent Boolean logic:



Intersection



Union

- Important in 3-D modeling because Boolean operations permit creation of complex objects from relatively simple ones

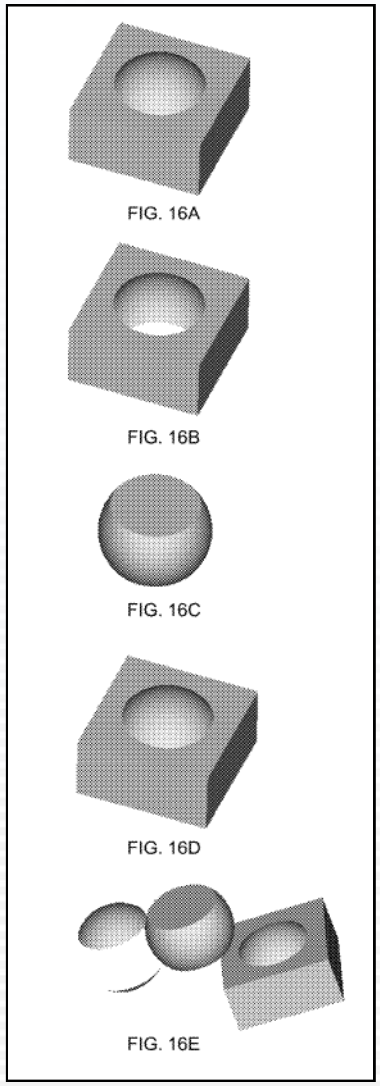
Boolean Operations in the Asserted Patents

Patent "presents five (5) kinds of Boolean operations: combination, intersection, exclusion, difference, and division"

(*See* '961 Patent at 8:9-60.)

Figure 16 illustrates Boolean operations on a box and a sphere

(*See* '961 patent at 3:49-52, Figs. 16A, 16B.)



Combination (Union)

Difference

Intersection

Exclusion

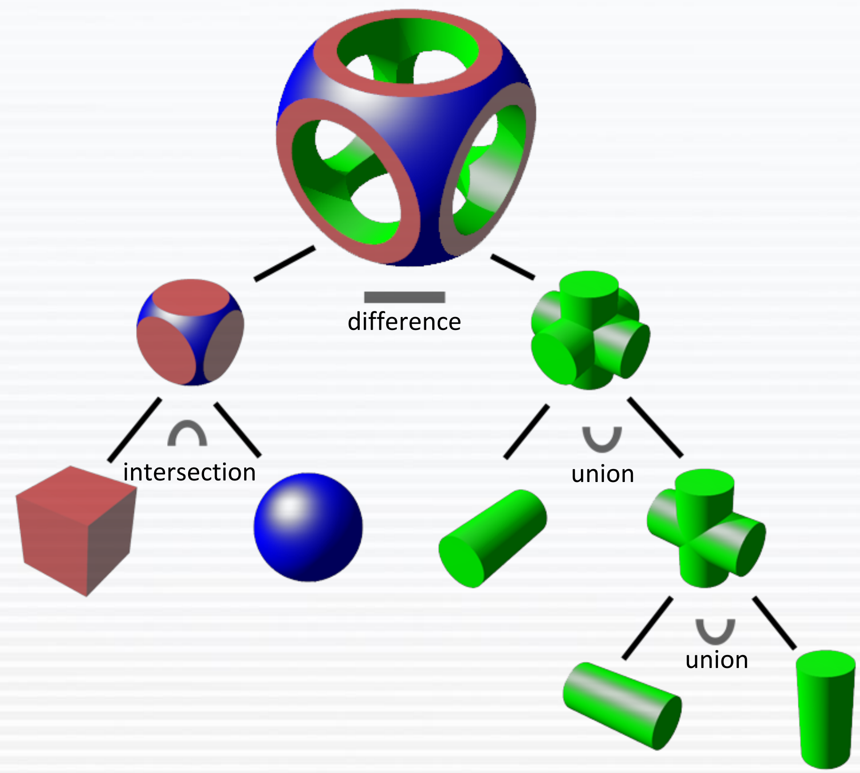
Division

Contents

- 3-D Modeling Overview
- Technical Concepts Used in 3-D Modeling
  - Boolean Operations
  - Constructive Solid Geometry
  - Boundary Representation
  - Facets
  - Triangulation
- Other Concepts in Patents
- Autodesk 3ds Max

Constructive Solid Geometry

- CSG is a volumetric approach to 3-D modeling
- Builds complex objects from simple ones (e.g., spheres or cubes)

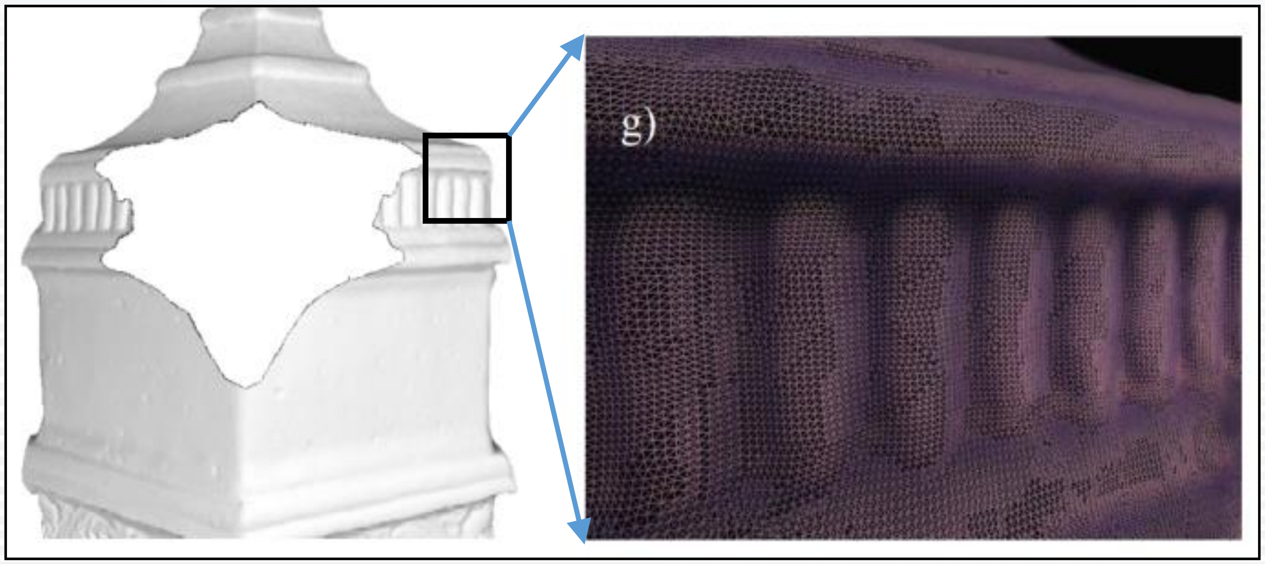


https://en.wikipedia.org/wiki/Constructive_solid_geometry (labels added)

Contents

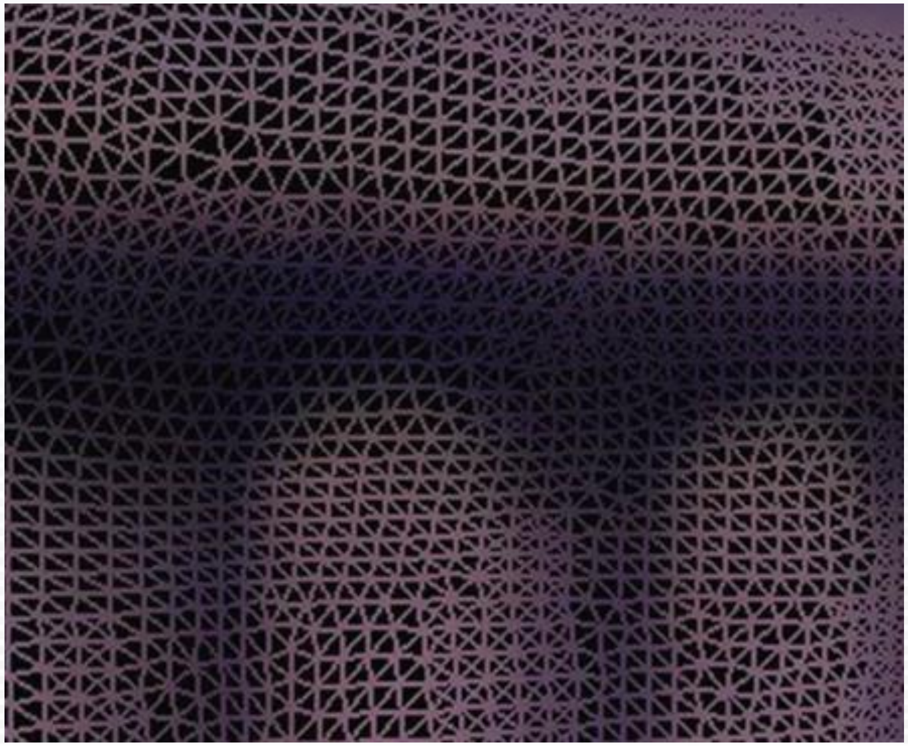
Boundary Representation

- B-Rep: Representation of the surface of a solid object
- Typically "closed" and supports "walking over" surface
- Can be defined by implicit surfaces, curved patches, triangle meshes, quadrilateral meshes, or hexagonal meshes



[Aliaga 2008]

Boundary Representation



[Aliaga 2008]

Contents

- 3-D Modeling Overview
- Technical Concepts Used in 3-D Modeling
  - Boolean Operations
  - Constructive Solid Geometry
  - Boundary Representation
  - Facets
  - Triangulation
- Other Concepts in Patents
- Autodesk 3ds Max

Facets

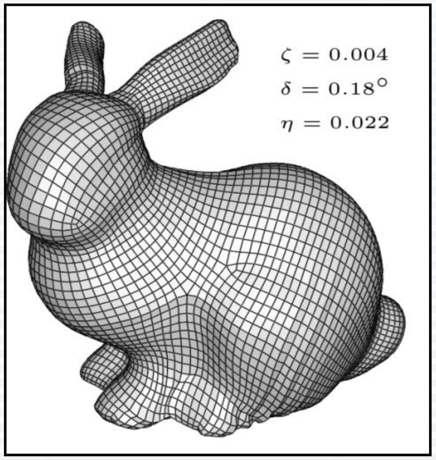
(57) **ABSTRACT**

A method for performing Boolean operations using a computer to create geometric models from primary geometric objects and their facets, comprises mapping rendering facets to extended triangles that contain neighbors; building intersection lines, splitting each triangle through which an intersection line passes, determining each facet is visible or obscure, and regrouping the facets to form one or more geometric objects. This method does not utilize the most popular data structures CSG and B-REP in CAD/CG/Solid Modeling systems, but has the advantages of both CSG and B-REP: easy to implement and flexible. Additionally it is a united method for solid modeling and surface modeling systems, and it is able to generate variant and editable models.

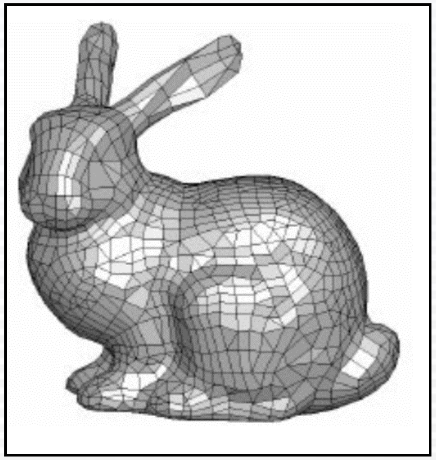
('961 Patent at Abstract.)

Facets

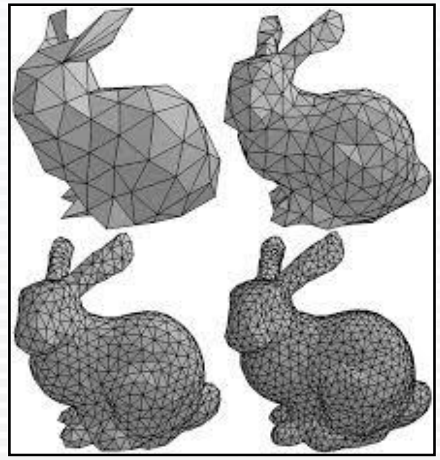
- Polygons that make up surface representation of a 3-D object
- Possible to represent 3-D object with different types and numbers of facets



quadrilaterals
[Levy et al. 2010]



triangles and quadrilaterals
[Liu et al. 2016]



varying number of triangles
[Luebke et al. 2002]

Contents

- 3-D Modeling Overview
- Technical Concepts Used in 3-D Modeling
  - Boolean Operations
  - Constructive Solid Geometry
  - Boundary Representation
  - Facets
  - Triangulation
- Other Concepts in Patents
- Autodesk 3ds Max

Triangles

Triangles are practical for 3-D modeling and mathematical operations

(57) **ABSTRACT**

A method for performing Boolean operations using a computer to create geometric models from primary geometric objects and their facets, comprises mapping rendering facets to extended triangles that contain neighbors; building intersection lines, splitting each triangle through which an intersection line passes, determining each facet is visible or obscure, and regrouping the facets to form one or more geometric objects. This method does not utilize the most popular data structures CSG and B-REP in CAD/CG/Solid Modeling systems, but has the advantages of both CSG and B-REP: easy to implement and flexible. Additionally it is a united method for solid modeling and surface modeling systems, and it is able to generate variant and editable models.

('961 Patent at Abstract.)

Triangulation

- Types of triangulation:
  - Delaunay
  - Fan-Based
  - Hamiltonian
- Methods of triangulation:
  - Random flipping
  - Watson algorithm
  - Ear clipping

Different Triangulations Have Different Properties

- Some triangulations produce “nicely shaped” triangles (e.g. good aspect ratio)
- Some triangulations have low calculation cost
- Some triangulations enable faster rendering computation

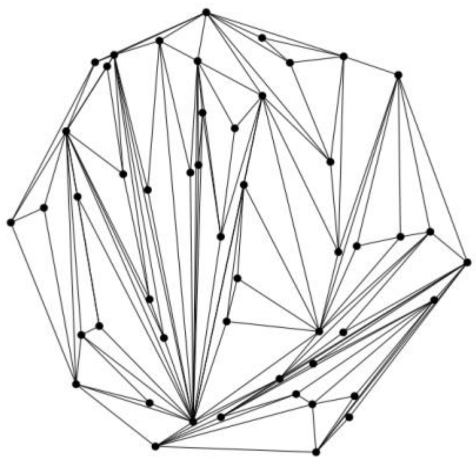

(a) Scan triangulation. (b) Delaunay triangulation.

Figure 6.3: *Two triangulations of the same set of 50 points.*

[Gartner et al. 2013]

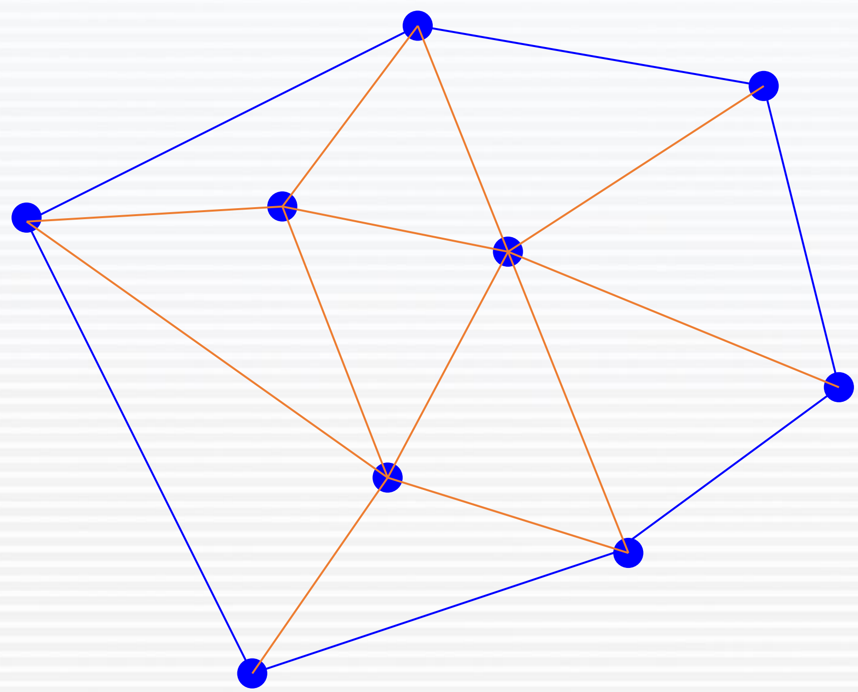
Triangulation

- Types of triangulation:
  - **Delaunay**
  - Fan-Based
  - Hamiltonian

- Methods of triangulation:
  - Random flipping
  - Watson algorithm
  - Ear clipping

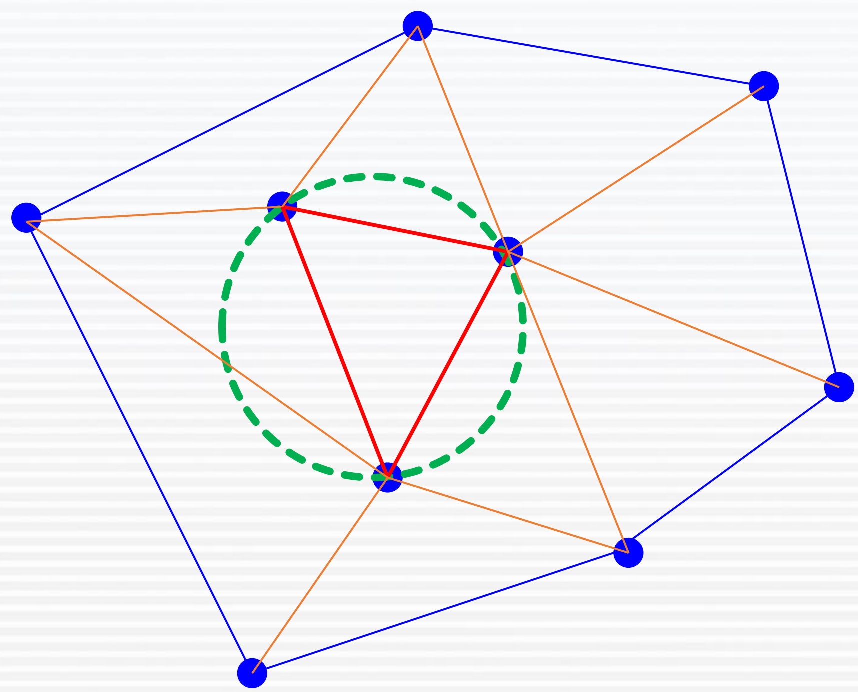
Delaunay Triangulation

Delaunay triangulations are a type of triangulation that produces “nicely shaped” triangles



Delaunay Triangulation

- Defining property: No triangle's circumcircle contains any of the points in the triangulation
- A circumcircle is the circle that contains the three points of a triangle

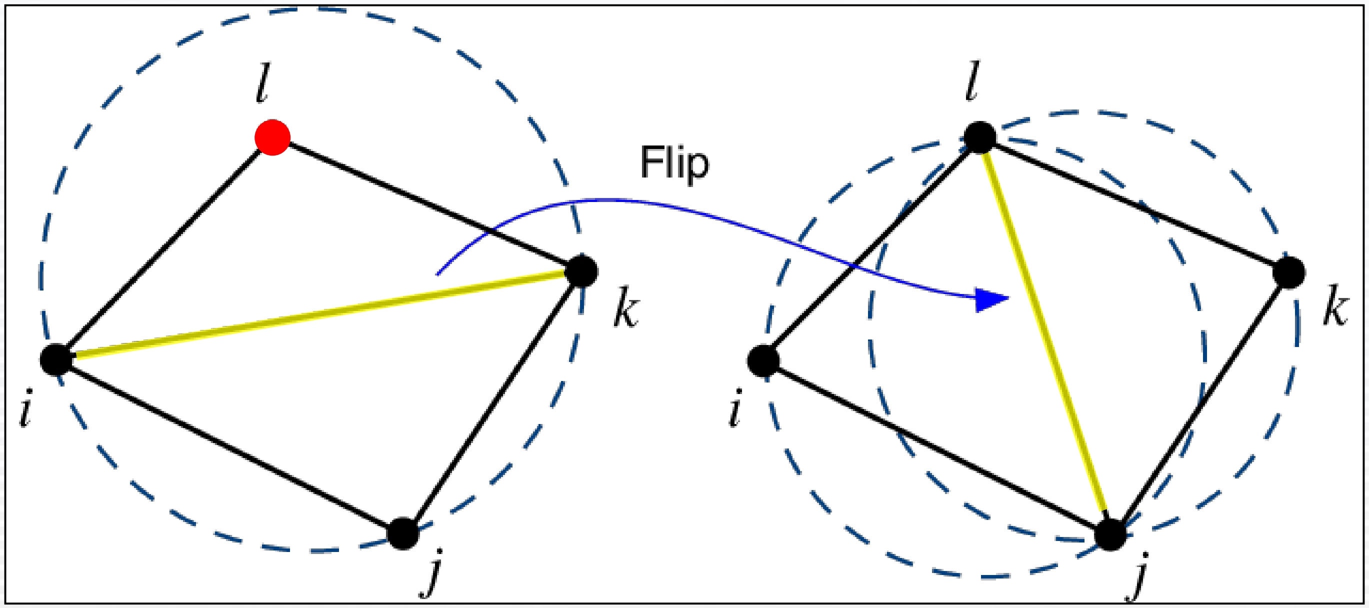


Triangulation

- Types of triangulation:
  - Delaunay
  - Fan-Based
  - Hamiltonian
- Methods of triangulation:
  - **Random flipping (Delaunay)**
  - Watson algorithm (Delaunay)
  - Ear clipping (non-Delaunay)

Delaunay Triangulation: Random Flipping

- Start with any triangulation
- Randomly flip edges until none of the points are inside any triangle's circumcircle



Non-Delaunay:
point *l* is inside the circumcircle of the triangle *ijk*

Delaunay:
no point is inside the circumcircle of either triangle

Triangulation

- Types of triangulation:
  - Delaunay
  - Fan-Based
  - Hamiltonian

- Methods of triangulation:
  - Random flipping (Delaunay)
  - **Watson algorithm (Delaunay)**
  - Ear clipping (non-Delaunay)

Delaunay Triangulation: Watson (1981)

Other Publications

"Non-Regularlized Boolean Set Operations on Non-Manifold B-Rep Objects", E. Gursoz et al., Carnegie Mellon University, Technical Report, 1990.

"Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes", D. F. Watson, The Computer Journal 24 (2) 1981.

('961 Patent at 9:10-15.)

Delaunay Triangulation: Watson (1981)

Computing the *n*-dimensional Delaunay tessellation with application to Voronoi polytopes†

D. F. Watson

Department of Geology and Geophysics, Edgeworth-David Building, University of Sydney, NSW 2006, Australia

The Delaunay tessellation in *n*-dimensional space is a space-filling aggregate of *n*-simplices. These *n*-simplices are the dual forms of the vertices in the commonly used Voronoi tessellation. Several efforts have been made to simulate the 2-dimensional Voronoi tessellation on the computer. Additional problems occur for the 3 and higher dimensional implementations but some of these can be avoided by alternatively computing the dual Delaunay tessellation. An algorithm that finds the topological relationships in these tessellations is given.

(Received October 1979, revised July 1980)

1. Introduction

The Delaunay tessellation, in 3-dimensional space, is an aggregate of space-filling, disjoint, irregular tetrahedra and may be thought of as a particular 3-dimensional triangulation. In 2-dimensional space it is an array of triangles. Such a triangulation provides an essential step for spatial analysis of punctulate data. Contouring programs usually require information concerning the degree of consanguinity amongst the data points, such as a grid or triangulation, before comparisons of the functional values at these data points are meaningful.

The Delaunay triangulation is also the geometric dual of the Voronoi tessellation. This tessellation has been used as a model in many areas of applied science. In particular, it is assumed to approximate the interface network in polycrystalline materials. To use it in this manner, the Voronoi tessellation must be considered to be a finite sample taken from an infinite population of which it is typical. However, in this respect, its form is not well known in ⩾ 3-dimensional space.

This paper presents an algorithm for computing the structure of the Delaunay *n*-dimensional triangulation for any stochastic array of data points. Such an algorithm can be used to find all of the adjacency relationships amongst N data points by triangulation in *n*-dimensions and in time of $O(N^{[(2n-1)/n]})$, provided that the N data points are partially ordered.

Further, the structure of the dual Voronoi tessellation is immediately obtained by geometric conversion. However, only those Voronoi domains strictly interior to the Delaunay tessellation are available since the boundary of the Delaunay tessellation yields only partial information on the Voronoi structure used as a polycrystalline model.

1.1 *The Delaunay Structure*

The defining points, or nuclei, of the 3-dimensional Delaunay tessellation lie at the vertices of the tetrahedra. Several tetrahedra will share an edge and even more will share a vertex. The four vertices of each tetrahedron lie on the surface of a sphere and no other vertex of the array lies within that sphere (see Miles, 1970, p. 107). This is because the centre of the sphere is in the position of a vertex in the dual Voronoi tessellation, and this vertex is by definition equidistant from the four nuclei.

This description for 3-dimensional space is readily extended to *n*-dimensional space. Thus the tetrahedron, or 3-simplex, becomes an *n*-simplex. Likewise, the $n + 1$ vertices of each *n*-simplex lie on the $(n - 1)$-dimensional surface of an *n*-dimensional hypersphere. The centre of that hypersphere, the simplicial circumcentre, is a vertex in the dual *n*-dimensional Voronoi tessellation. In all cases the *n*-simplices of a Delaunay tessellation are space-filling and disjoint, while for stochastic arrays of defining nuclei they are also uniquely defined.

1.2 *The Voronoi Structure*

A Voronoi tessellation partitions *n*-dimensional space into convex polytopes and may be thought of as a network of interfaces formed by impingement of expanding hyperspheres, centred at the nuclei, and growing at a constant rate from time zero. Thus some $(n + 1)$-tuples of such hyperspheres meet a mutual conclusion at a point which is equidistant from the $n + 1$ nuclei. That point is a vertex in the Voronoi tessellation. Therefore these $(n + 1)$-tuples of nuclei may be used to define a vertex in a Voronoi tessellation, or equivalently, an *n*-simplex in a Delaunay tessellation.

According to Sommerville (1927, p. 106), the numbers of *i*-dimensional domains on an *n*-simplex is ${}_{n+1}C_{i+1}$, for $0 \leqslant i \leqslant n$. Then, since the geometrical dual in *n*-dimensional space of an *i*-dimensional domain is an $(n - i)$-dimensional domain, we know the numbers of $(n - i)$-dimensional domains that include a 0-dimensional domain, or vertex, in a Voronoi tessellation is ${}_{n+1}C_{n-i}$. This duality means that the total of *n*-simplices incident to a given nucleus in a Delaunay tessellation is equal to the number of vertices on the dual Voronoi domain associated with that nucleus. The total of *n*-simplices in common with a given neighbouring nucleus is the number of vertices on the $(n - 1)$-dimensional face formed with that neighbour in the dual Voronoi domains. Similar remarks apply to other such values.

1.3 *Previous Work*

The theoretical properties of the Voronoi planar tessellation have been established by Miles (1970, p. 103). A computer simulation, of the space-filling members of an aggregate in the plane, was accomplished by Crain (1972). He generated individual polygons and tabulated their measurements to form histograms and estimate ergodic moments.

Various algorithms have been used to find the boundaries of the Voronoi polygons. Knowing that the nearest neighbouring nucleus to some given nucleus must form an interface, one may proceed by tracing a pathway, always in the same rotational direction, around each polygon, as was done by Rhynsburger (1973, p. 141). A more sophisticated approach by Green and Sibson (1978, p. 170), traces the boundary adjustments required as a new nucleus, and thus a new polygon, is fitted into an established aggregate.

Perimeter tracing, however, though a principal technique for the plane, does not adapt to ⩾ 3-dimensional space. Calling on the mathematical definition of the Voronoi domain as an intersection of half-spaces (see Rogers, 1964, p. 74), the 3-dimen-

†Editorial note: This paper and that by Bowyer (this issue) cover some material in common. As these contributions were received at approximately the same time, the Editor feels it only right to include both papers.

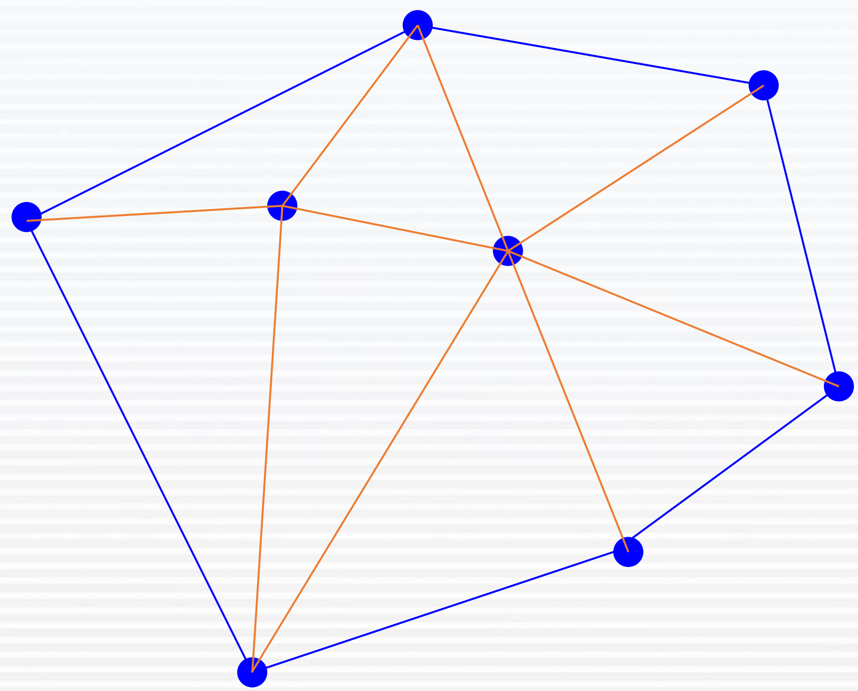
CCC-0010-4620/81/0024-0167 $03.00

Incrementally computes a Delaunay Triangulation:

1. Start with an initial Delaunay Triangulation
2. Add a point
3. Identify triangles whose circumcircles contain the added point
4. Remove those triangles
5. Triangulate the polygon formed by the empty space left behind

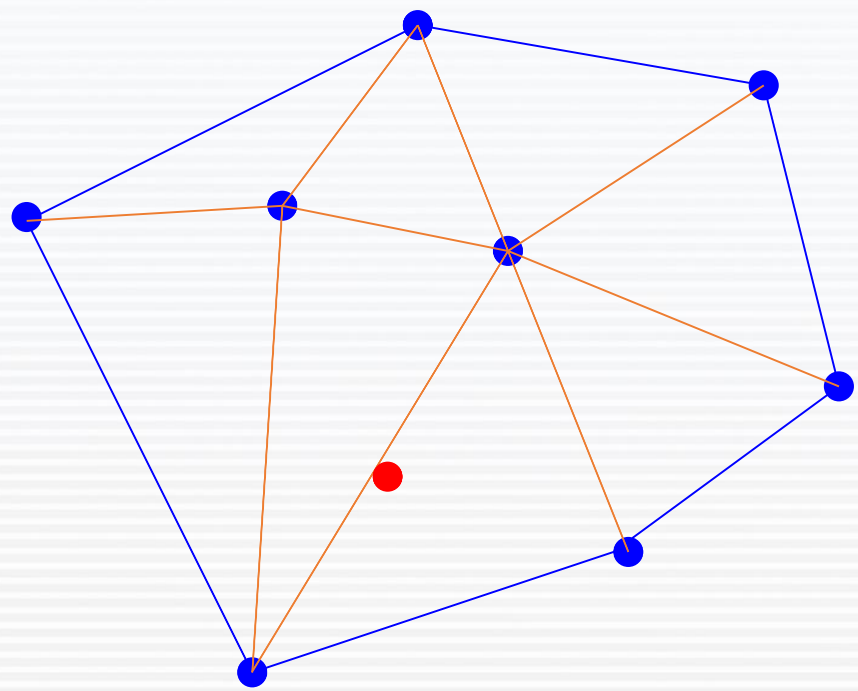
Delaunay Triangulation: Watson (1981)

1. Start with an initial Delaunay Triangulation:



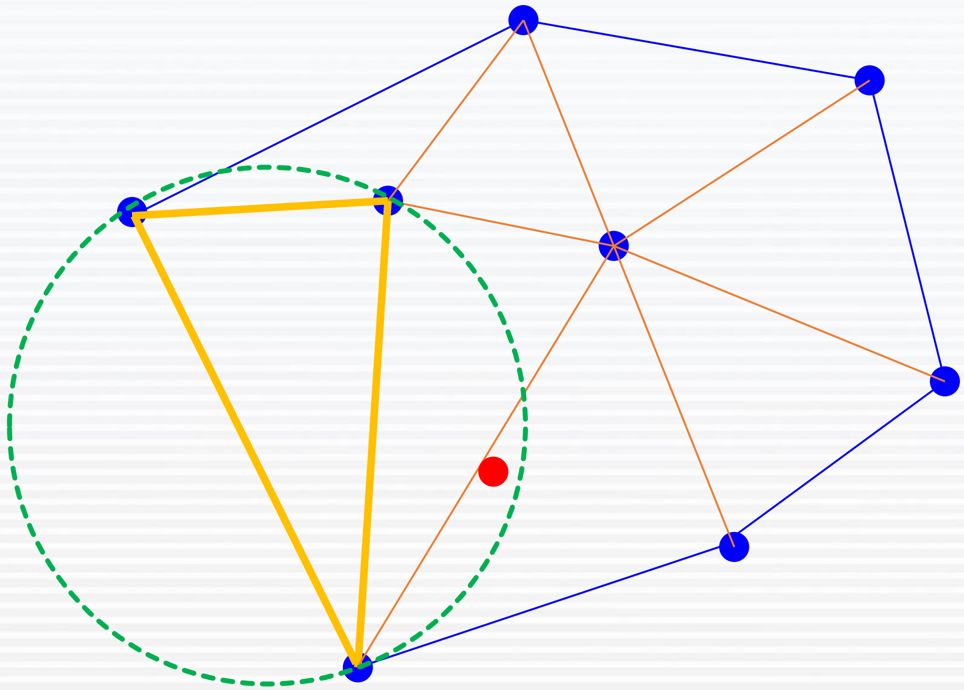
Delaunay Triangulation: Watson (1981)

2. Add **a point**:



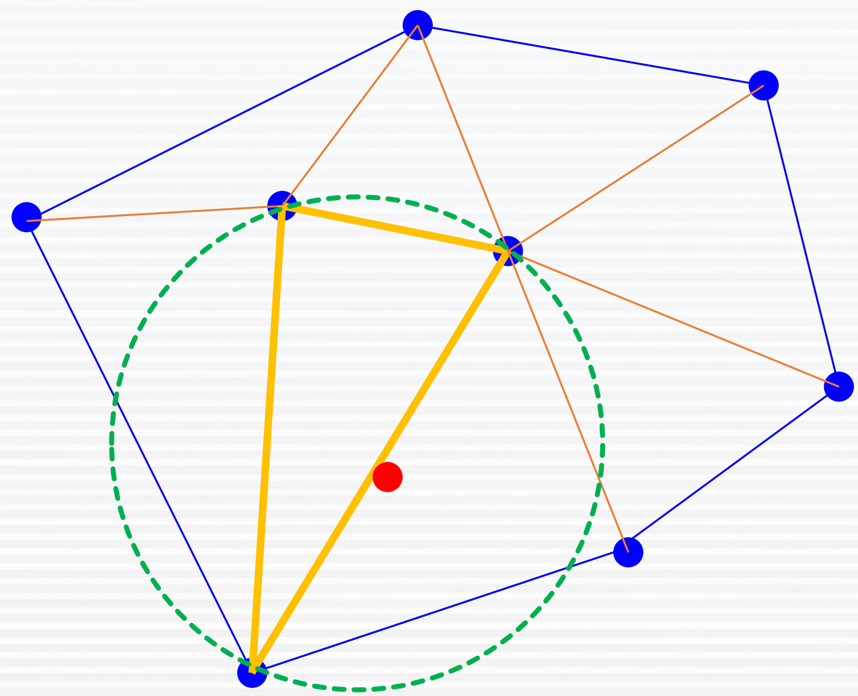
Delaunay Triangulation: Watson (1981)

3. Identify **triangles** whose **circumcircles** contain the added **point**:



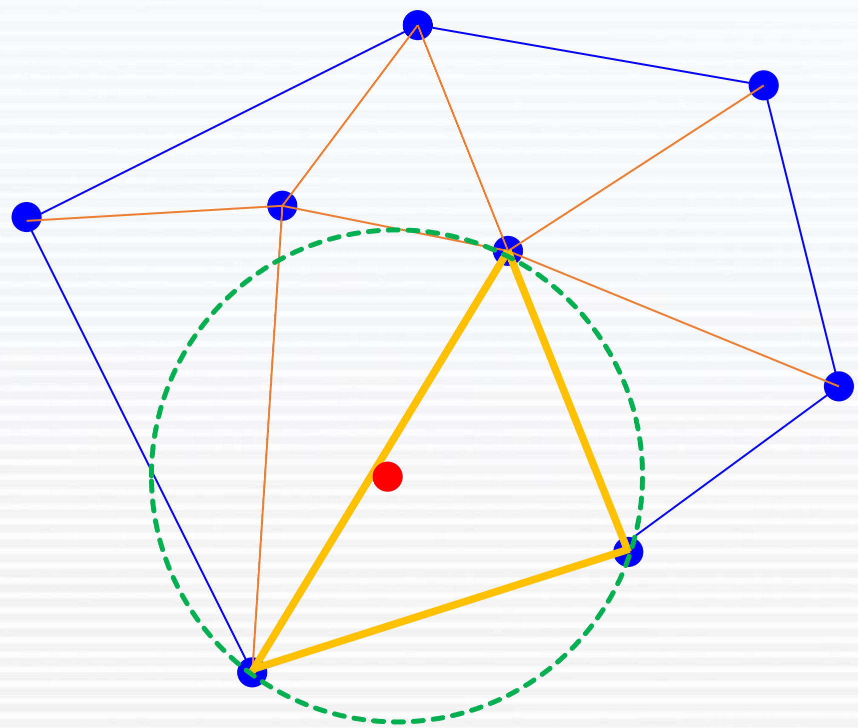
Delaunay Triangulation: Watson (1981)

3. Identify **triangles** whose **circumcircles** contain the added **point**:



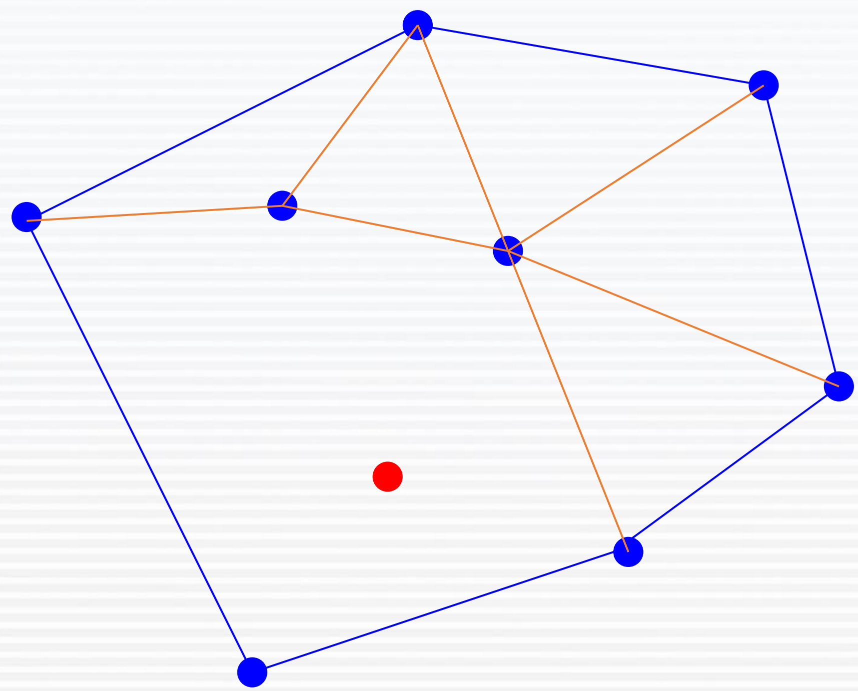
Delaunay Triangulation: Watson (1981)

3. Identify triangles whose circumcircles contain the added point:



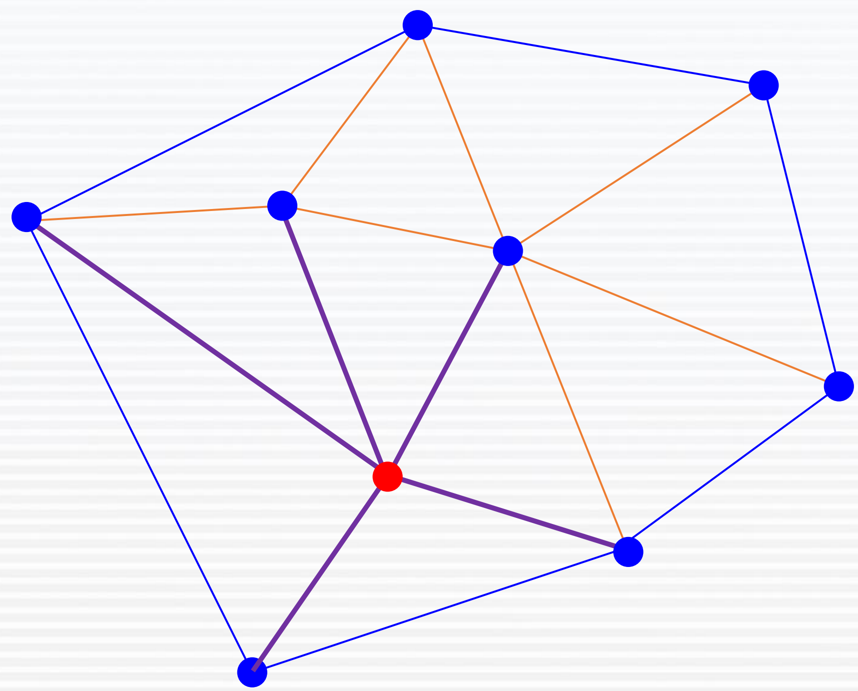
Delaunay Triangulation: Watson (1981)

4. Remove the identified triangles:



Delaunay Triangulation: Watson (1981)

5. Triangulate the polygon formed by the empty space left behind:



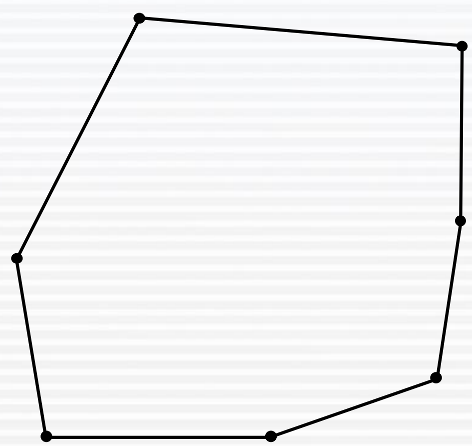
Triangulation

- Types of triangulation:
  - Delaunay
  - Fan-Based
  - Hamiltonian

- Methods of triangulation:
  - Random flipping (Delaunay)
  - Watson algorithm (Delaunay)
  - **Ear clipping (non-Delaunay)**

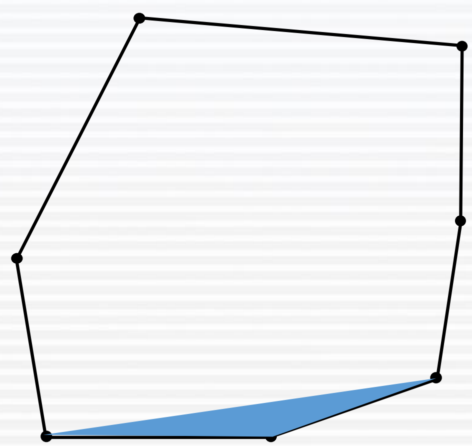
Non-Delaunay Triangulation: Ear Clipping

- Clipping away “ears” of polygon to form triangles
- Algorithm proceeds iteratively until polygon is divided into triangles
- Not for producing Delaunay triangulation



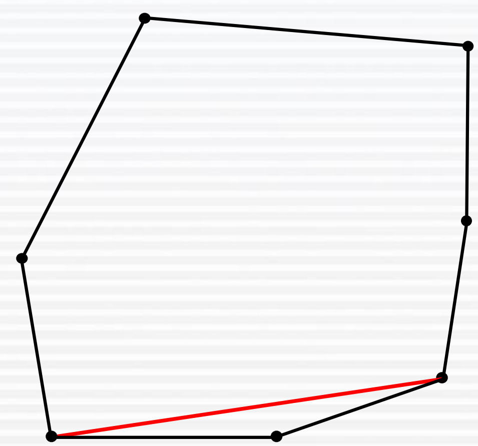
Non-Delaunay Triangulation: Ear Clipping

- Clipping away "ears" of polygon to form triangles
- Algorithm proceeds iteratively until polygon is divided into triangles
- Not for producing Delaunay triangulation



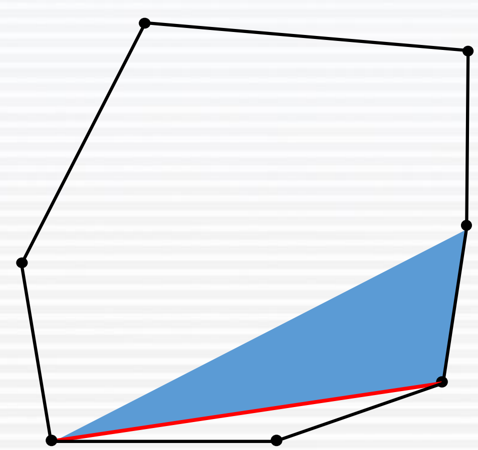
Non-Delaunay Triangulation: Ear Clipping

- Clipping away “ears” of polygon to form triangles
- Algorithm proceeds iteratively until polygon is divided into triangles
- Not for producing Delaunay triangulation



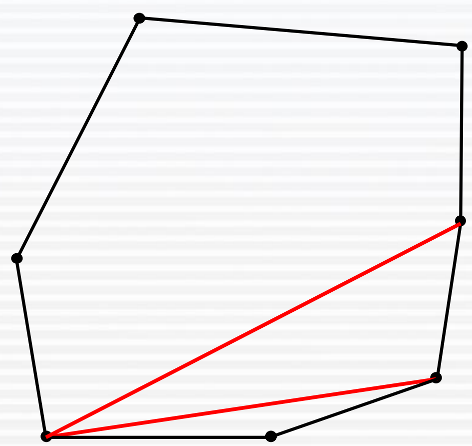
Non-Delaunay Triangulation: Ear Clipping

- Clipping away "ears" of polygon to form triangles
- Algorithm proceeds iteratively until polygon is divided into triangles
- Not for producing Delaunay triangulation



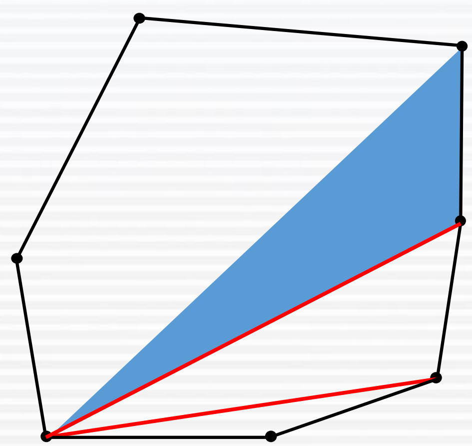
Non-Delaunay Triangulation: Ear Clipping

- Clipping away "ears" of polygon to form triangles
- Algorithm proceeds iteratively until polygon is divided into triangles
- Not for producing Delaunay triangulation



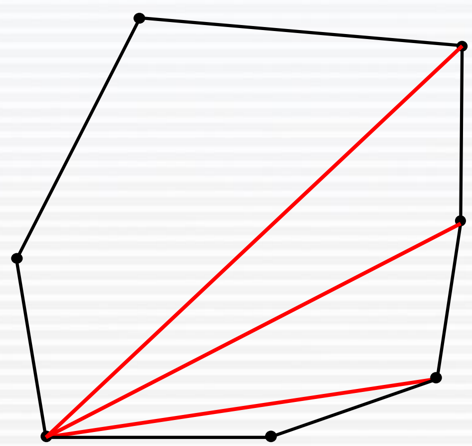
Non-Delaunay Triangulation: Ear Clipping

- Clipping away “ears” of polygon to form triangles
- Algorithm proceeds iteratively until polygon is divided into triangles
- Not for producing Delaunay triangulation



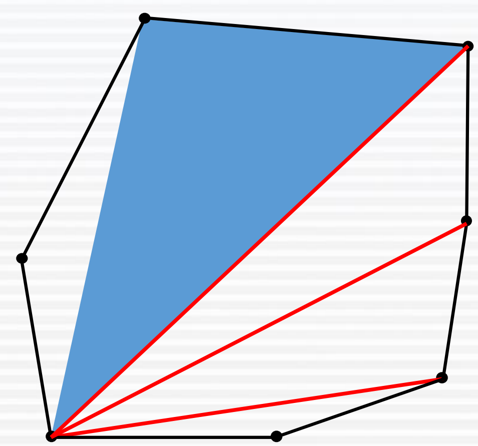
Non-Delaunay Triangulation: Ear Clipping

- Clipping away “ears” of polygon to form triangles
- Algorithm proceeds iteratively until polygon is divided into triangles
- Not for producing Delaunay triangulation



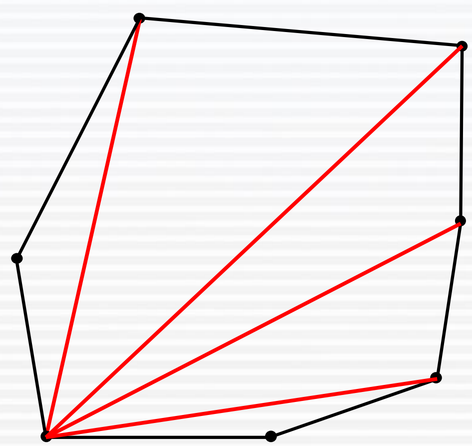
Non-Delaunay Triangulation: Ear Clipping

- Clipping away “ears” of polygon to form triangles
- Algorithm proceeds iteratively until polygon is divided into triangles
- Not for producing Delaunay triangulation



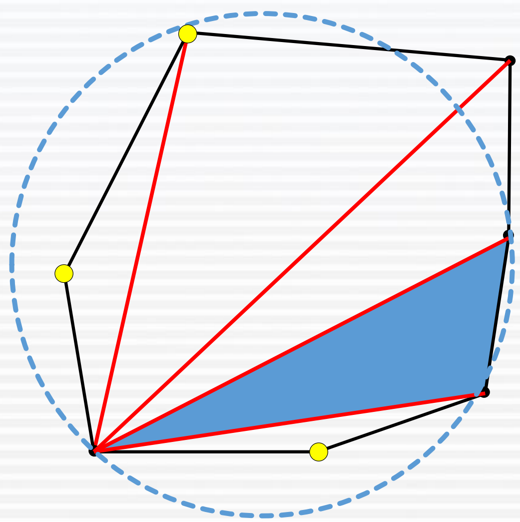
Non-Delaunay Triangulation: Ear Clipping

- Clipping away “ears” of polygon to form triangles
- Algorithm proceeds iteratively until polygon is divided into triangles
- Not for producing Delaunay triangulation



Non-Delaunay Triangulation: Ear Clipping

- Clipping away "ears" of polygon to form triangles
- Algorithm proceeds iteratively until polygon is divided into triangles
- Not for producing Delaunay triangulation



Contents

Triangle "Neighbors"

1. A method that performs immediate Boolean operations using geometric facets of geometric objects implemented in a computer system and operating with a computer, the method comprising:

mapping rendering facets to extended triangles that contain neighbors;

building intersection lines starting with and ending with searching for the first pair of triangles that hold a start

('961 Patent at 9:17-24.)

Triangle "Neighbors"

Remark 5: Every triangle has three (3) edges, when there are no duplicated points, it has three (3) neighboring triangles in solid models. FIG. **5** shows an example: a triangle filled with dark color and its three (3) neighbors. When concerning a surface patch for surface trimming, one or two neighbors of a triangle may be null.

('961 Patent at 3:5-7.)

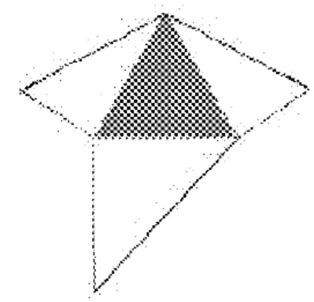


FIG. 5

Minimum Bounding Box

1. A method that performs immediate Boolean operations using geometric facets of geometric objects implemented in a computer system and operating with a computer, the method comprising:

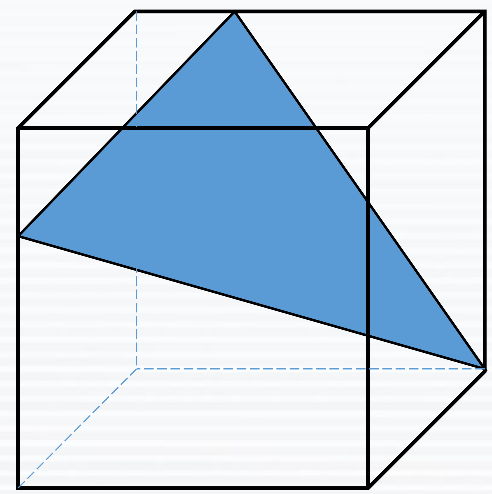
mapping rendering facets to extended triangles that contain neighbors;

building intersection lines starting with and ending with searching for the first pair of triangles that hold a start point of an intersection line by detecting whether two minimum bounding boxes overlap and performing edge-triangle intersection calculations for locating an intersection point, then searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed;

('961 Patent at 9:17-33.)

Minimum Bounding Box

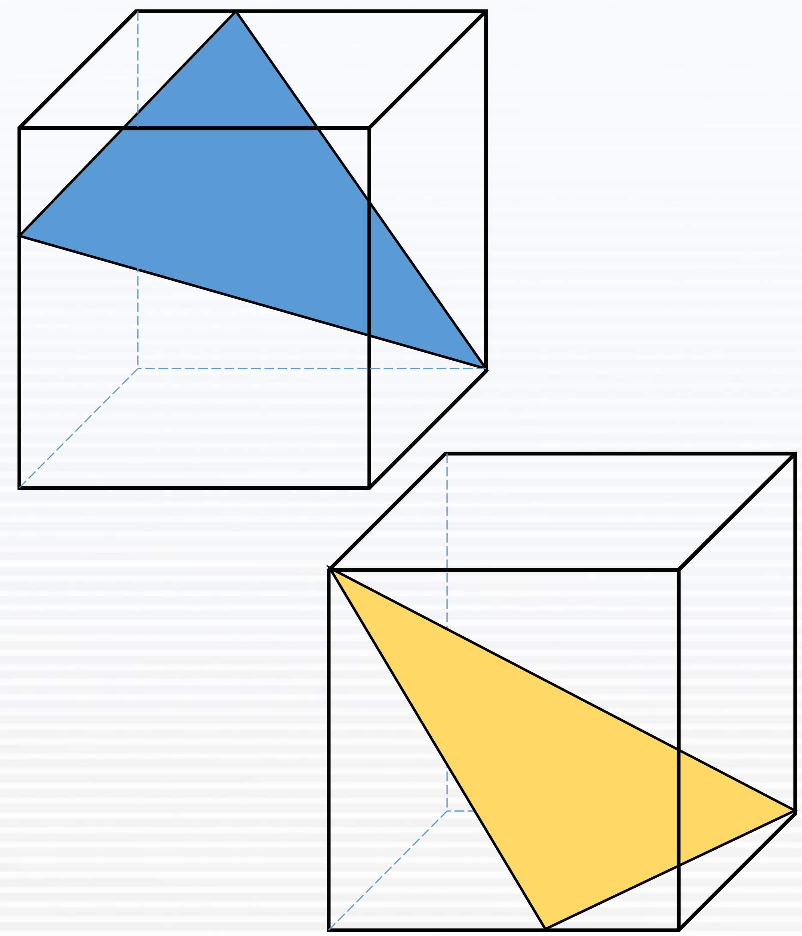
Smallest box enclosing all three vertices of a triangle



Minimum Bounding Box

- Computing whether boxes in 3-D space intersect is easier than computing whether triangles intersect

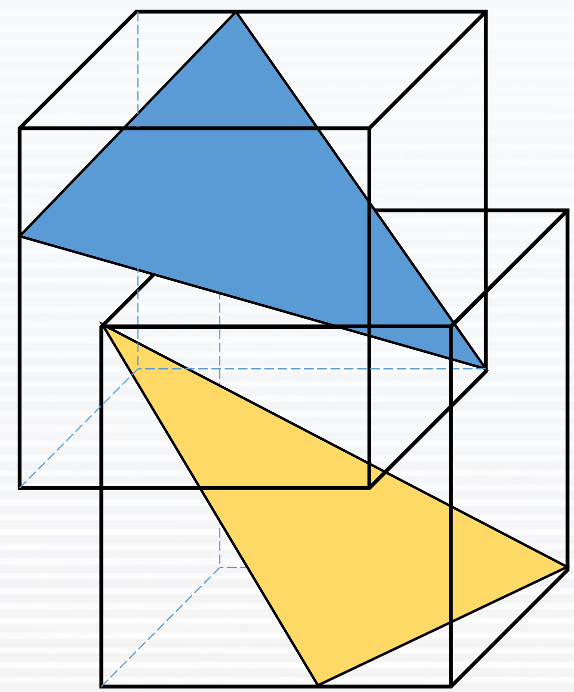
- Intersections
  - If boxes do not overlap, triangles do not overlap
  - If boxes overlap, need to do more work



Minimum Bounding Box

- Computing whether boxes in 3-D space intersect is easier than computing whether triangles intersect

- Intersections
  - If boxes do not overlap, triangles do not overlap
  - If boxes overlap, need to do more work



Minimum Bounding Box

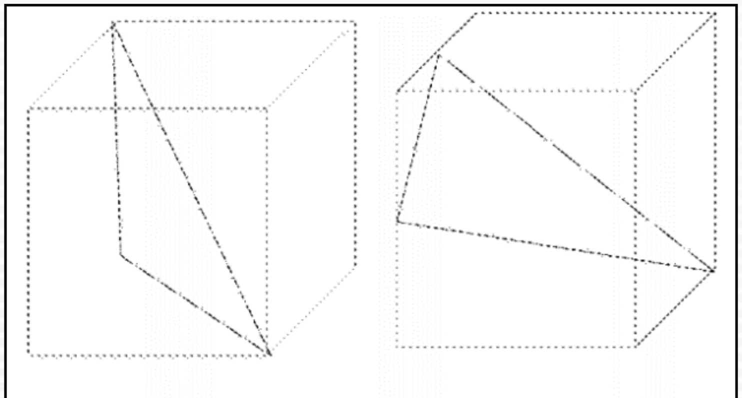


FIG. 6A

Non-Overlapping Boxes

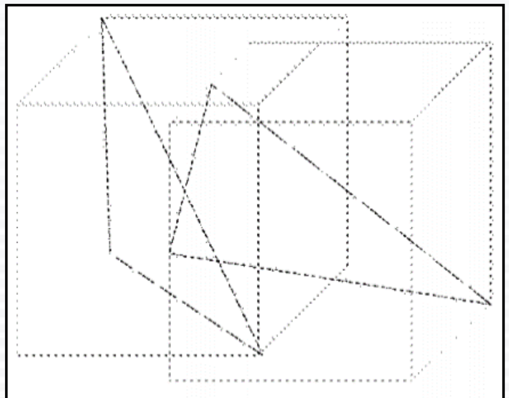


FIG. 6B

Overlapping Boxes

Surface Trimming

Surface trimming is a concept that appears in claim 1 of the ’961 patent, but not in claim 1 of the ’105 patent, which is otherwise identical.

1. A method that performs immediate Boolean operations using geometric facets of geometric objects implemented in a computer system and operating with a computer, the method comprising:

mapping rendering facets to extended triangles that contain neighbors;

building intersection lines starting with and ending with searching for the first pair of triangles that hold a start point of an intersection line by detecting whether two minimum bounding boxes overlap and performing edge-triangle intersection calculations for locating an intersection point, then searching neighboring triangles of the last triangle pair that holds the last intersection point to extend the intersection line until the first intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed;

splitting each triangle through which an intersection line passes using modified Watson method, wherein the modified Watson method includes removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion, and a middle portion;

checking each triangle whether it is obscure or visible for Boolean operations or for surface trimming;

regrouping facets in separate steps that includes copying triangles, deleting triangles, reversing the normal of each triangle of a geometric object, and merging reserved triangles to form one or more new extended triangle sets; and

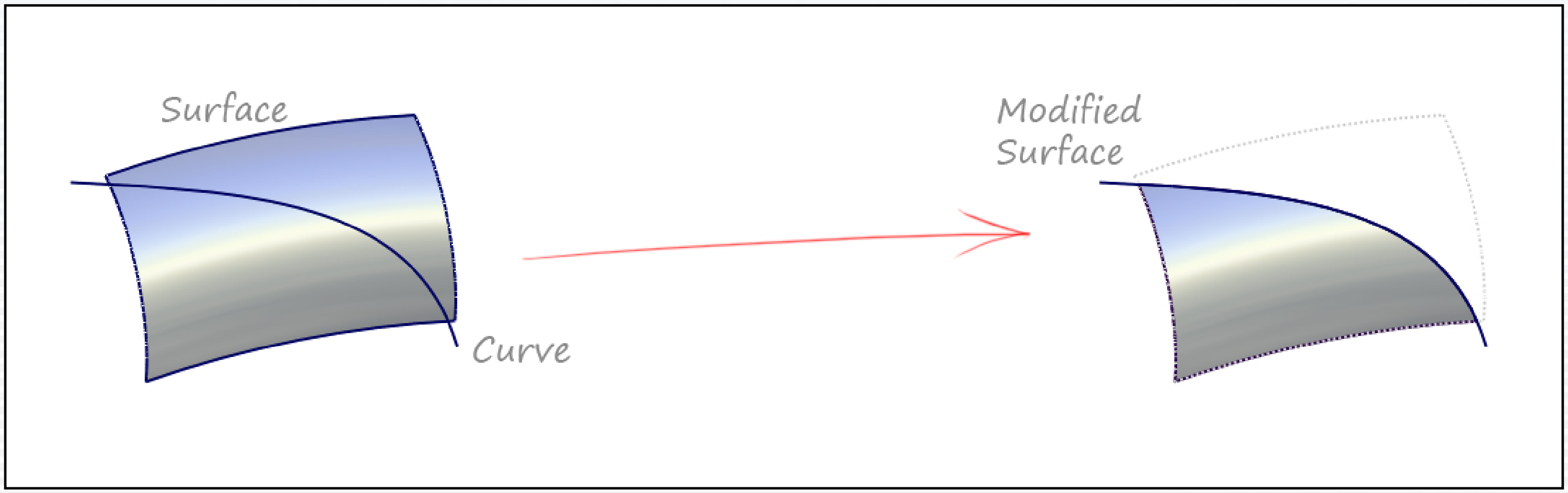
mapping extended triangles to rendering facets.

(’961 Patent at 9:17-48.)

Surface Trimming

Surface trimming:

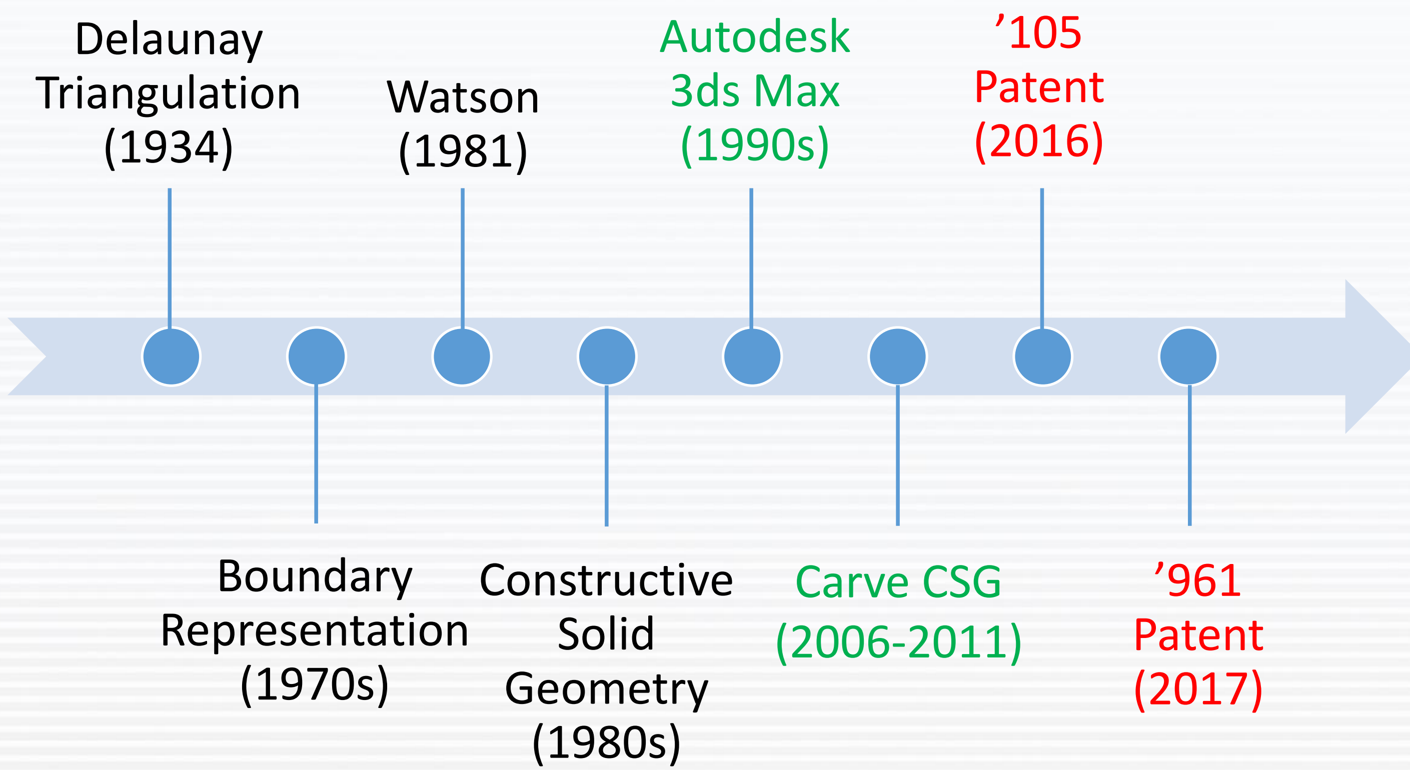
Trimming, or cutting, a surface along a curve or contour line



https://knowledge.autodesk.com/

Contents

History of 3-D Modeling

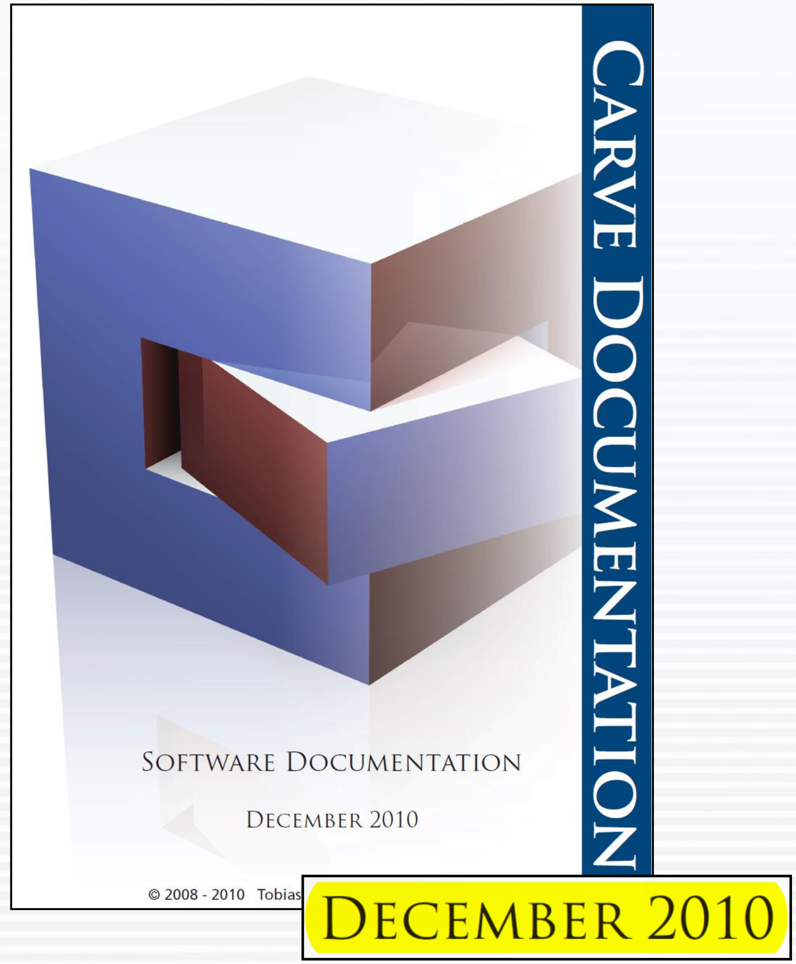


Autodesk 3ds Max

- Autodesk's 3ds Max software was developed in the 1990s
- 3ds Max allows users to model and render 3-D graphics for design visualization, games, and animation
- 3ds Max can perform Boolean operations and surface trimming, among other procedures

Carve CSG

- 3ds Max uses the Carve CSG Boolean library (as of April 2016)
- Carve CSG performs Boolean operations using polygons
- Carve CSG optionally triangulates by ear clipping
- Carve developed in mid-2000s



AUTODESK®

Nature Simulation Systems, Inc. v. Autodesk, Inc.

Case No. 3:19-cv-03192-SK

Rebuttal Slides

May 26, 2020

"Modified Watson Method?"

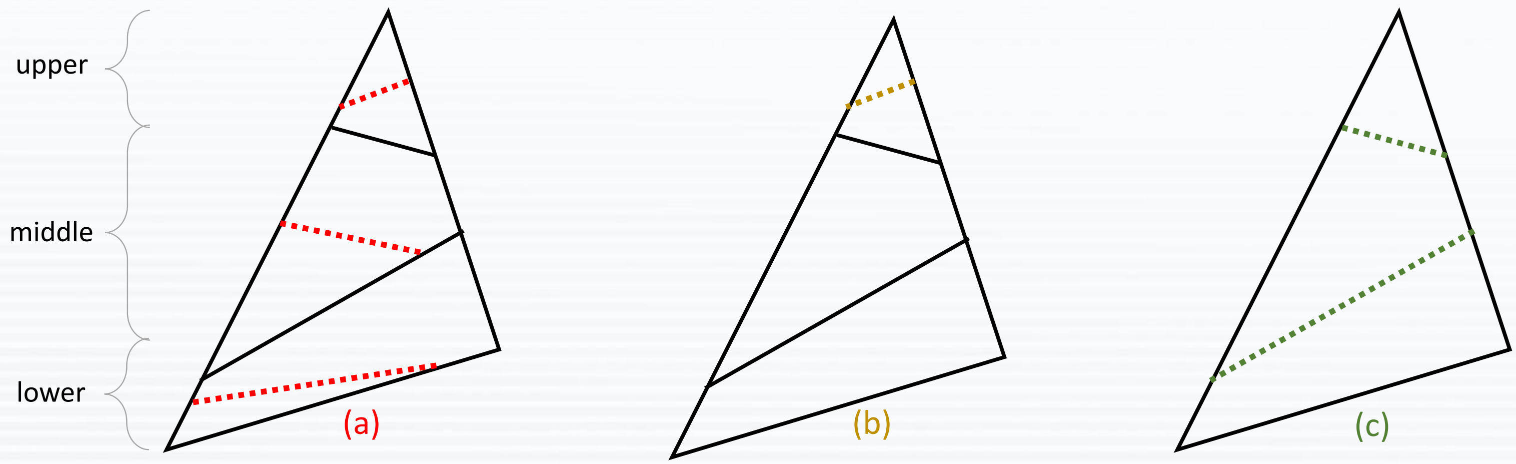
intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed;

splitting each triangle through which an intersection line passes using modified Watson method, wherein the modified Watson method includes removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion, and a middle portion;

checking each triangle whether it is obscure or visible for Boolean operations or for surface trimming;

('961 Patent at 9:31-41.)

"Modified Watson Method?"



(a) Splitting an upper portion, splitting a lower portion, and splitting a middle portion of each triangle;

(b) splitting one portion of each triangle, if that triangle has an upper portion, a lower portion, and a middle portion;

(c) splitting each triangle into an upper portion, a lower portion, and a middle portion; or

(d) Something else.

"Modified Watson Method?"

intersection point is identical to the last intersection point of the intersection line ensuring that the intersection line gets closed or until all triangles are traversed;

splitting each triangle through which an intersection line passes using modified Watson method, wherein the modified Watson method includes removing duplicate intersection points, identifying positions of end intersection points, and splitting portion of each triangle including an upper portion, a lower portion, and a middle portion;

checking each triangle whether it is obscure or visible for Boolean operations or for surface trimming;

('961 Patent at 9:31-41.)

"Modified Watson Method?"

FIG. **13** is the flowchart of Delaunay mesh modified Watson method that created the sequence of FIGS. **12**A through **12**H.

('961 Patent at 3:39-41.)

“Modified Watson Method?”

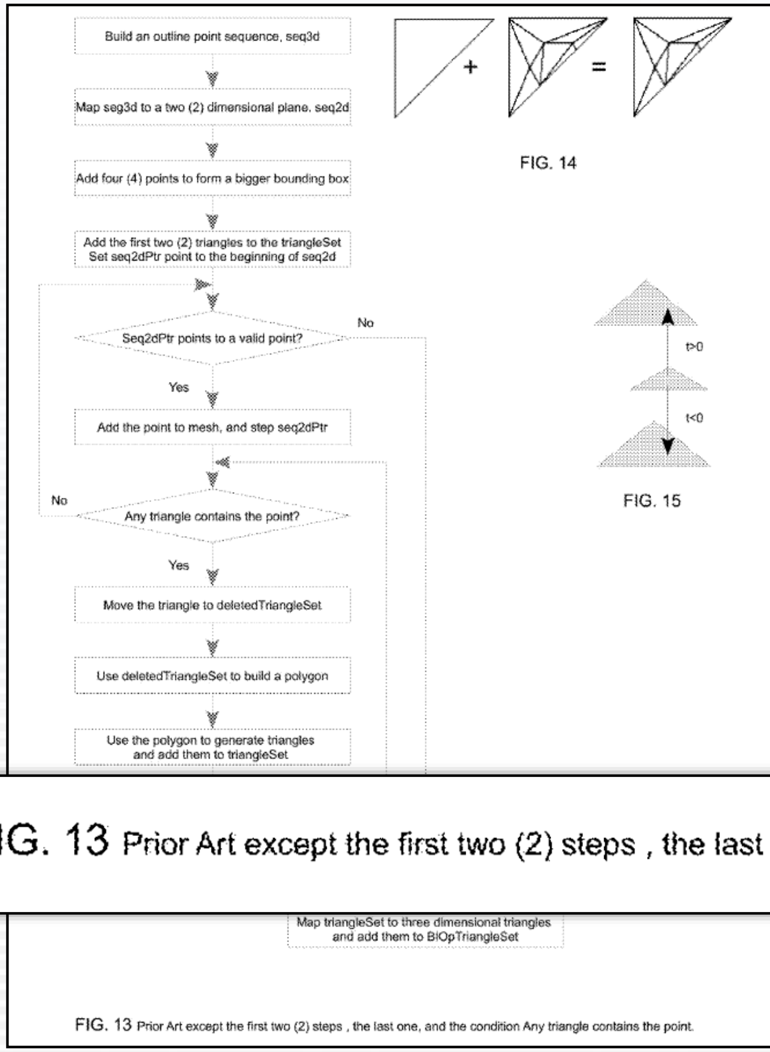
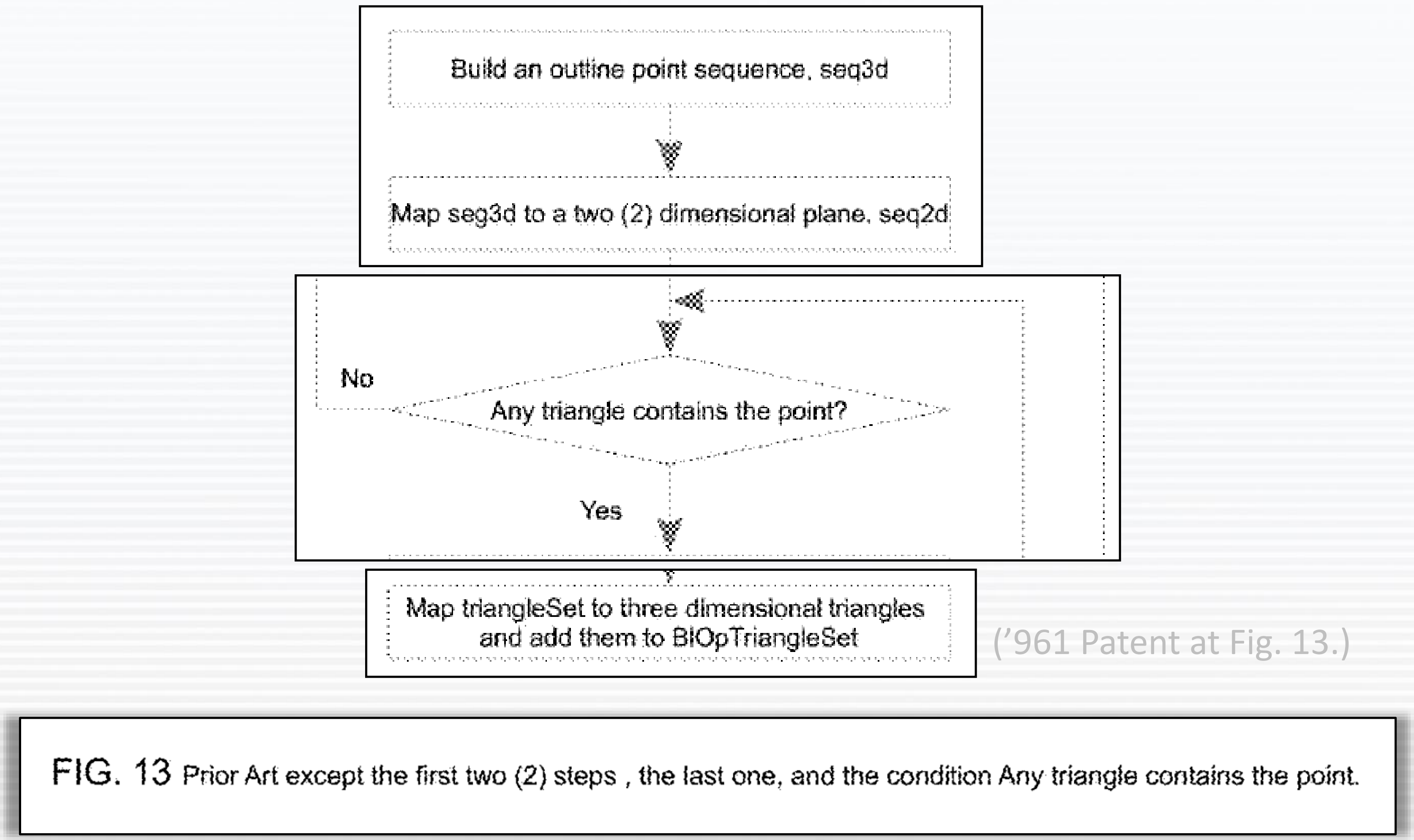


FIG. 13 Prior Art except the first two (2) steps , the last one, and the condition Any triangle contains the point.

FIG. 13 Prior Art except the first two (2) steps , the last one, and the condition Any triangle contains the point.

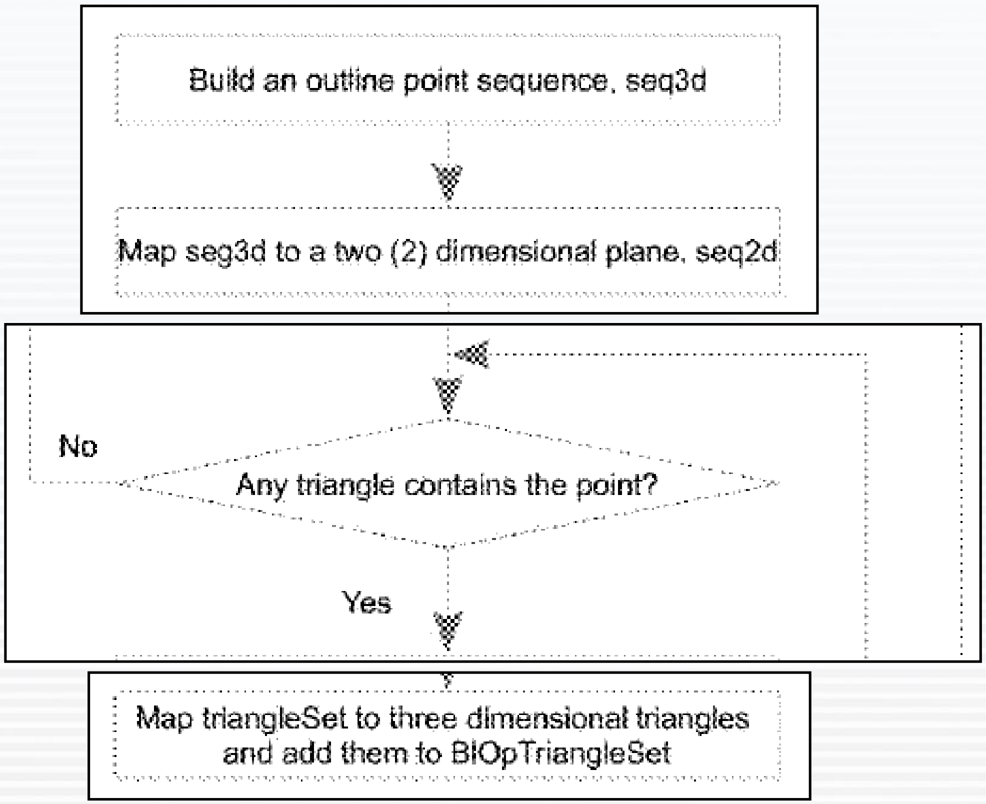
(’961 Patent at Fig. 13.)

"Modified Watson Method?"



('961 Patent at Fig. 13.)

"Modified Watson Method?"



('961 Patent at Fig. 13.)

"wherein the modified Watson method includes

- removing duplicate intersection points,
- identifying positions of end intersection points, and
- splitting portion of each triangle including an upper portion, a lower portion, and a middle portion"

('961 Patent at 9:34-40.)

AUTODESK®

Nature Simulation Systems, Inc. v. Autodesk, Inc.

Case No. 3:19-cv-03192-SK

Claim Construction Hearing

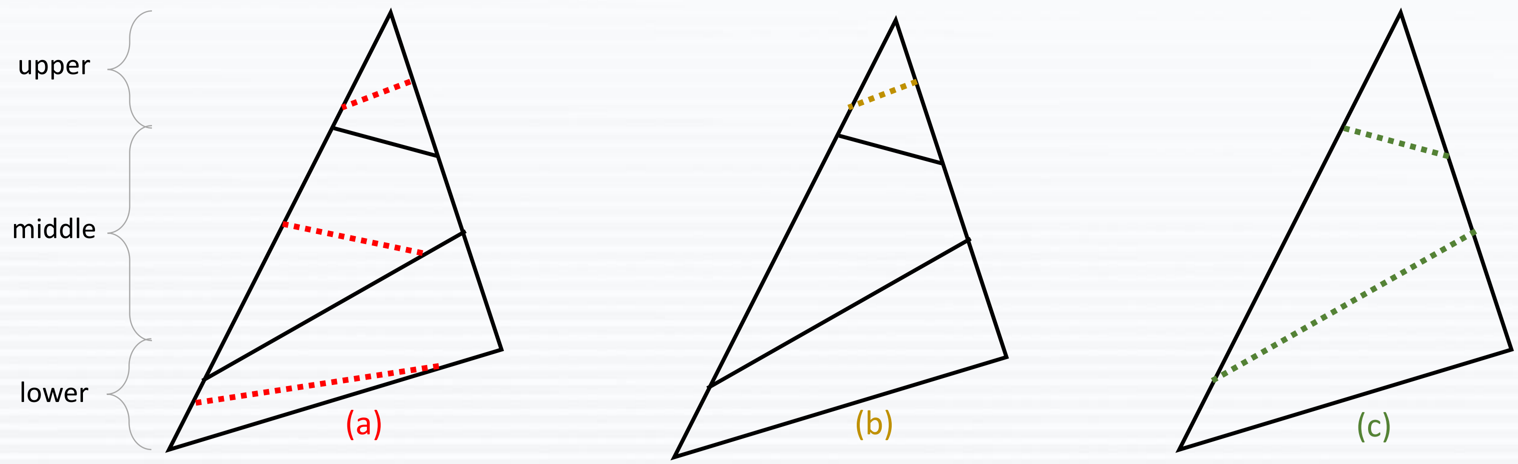
June 4, 2020

Contents

1. "modified Watson method"
   - Claim Language
   - Figure 13
   - Specification Text
   - Figure 12
   - Prosecution History
2. "searching neighboring triangles . . ."
   - Claim Language
   - Searching?
   - Iterative Process
3. "mapping rendering facets . . ."
4. "the intersection line gets closed . . ."
5. "regrouping facets in separate steps . . ."
6. "regular points"
7. "BIOpTriangleSet"
8. "according to m_ID . . . deciding . . ."

"modified Watson method"

"Splitting portion of each triangle . . ." is ambiguous.



(a) Splitting an upper portion, splitting a lower portion, and splitting a middle portion of each triangle;

(b) splitting one portion of each triangle, if that triangle has an upper portion, a lower portion, and a middle portion;

(c) splitting each triangle into an upper portion, a lower portion, and a middle portion; or

(d) Something else.

(*See* Aliaga Decl. ¶ 16.)

"modified Watson method"

The difference between Figures 12F and 12G is that a **polygon** (not a triangle) has been split.

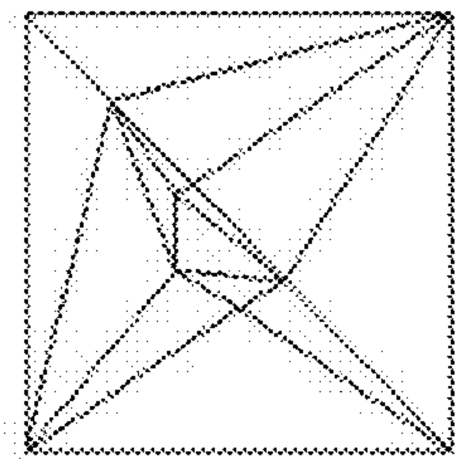


FIG. 12F Prior Art

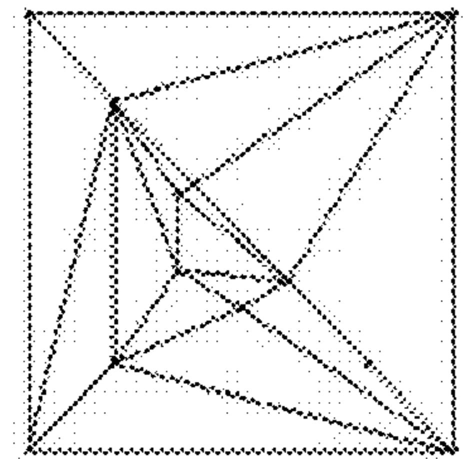


FIG. 12G

('961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12F and 12G is that a polygon (not a triangle) has been split.

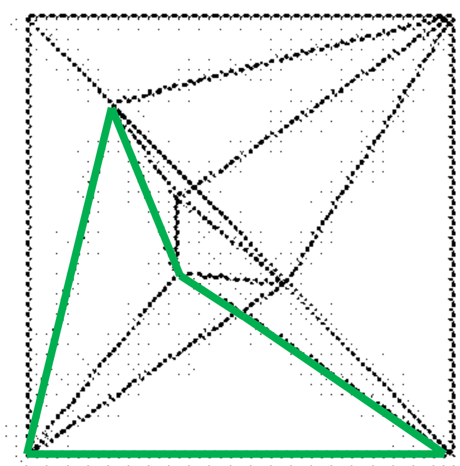


FIG. 12F Prior Art

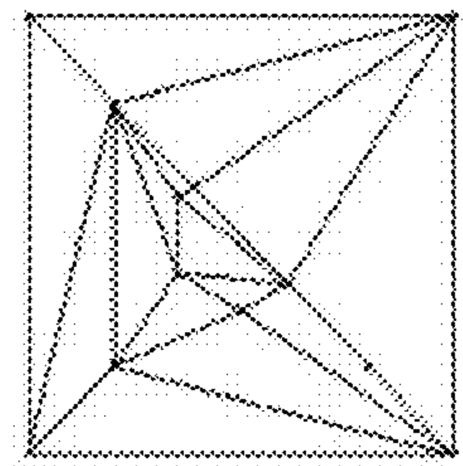


FIG. 12G

('961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12F and 12G is that a polygon (not a triangle) has been split.

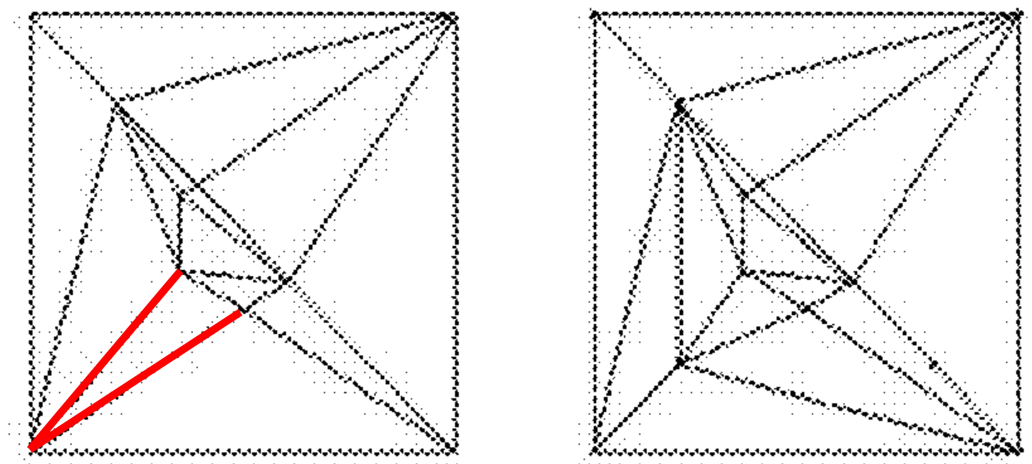


FIG. 12F Prior Art FIG. 12G

('961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12F and 12G is that a polygon (not a triangle) has been split.

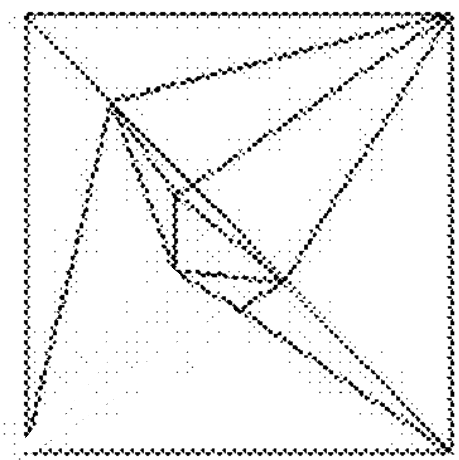


FIG. 12F Prior Art

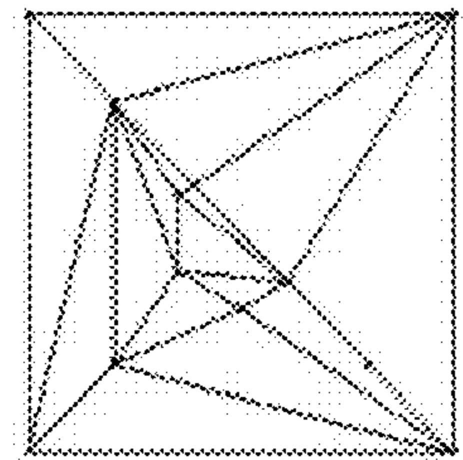


FIG. 12G

('961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12F and 12G is that a polygon (not a triangle) has been split.

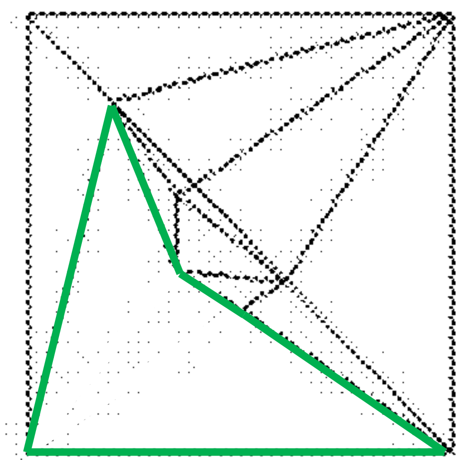


FIG. 12F Prior Art

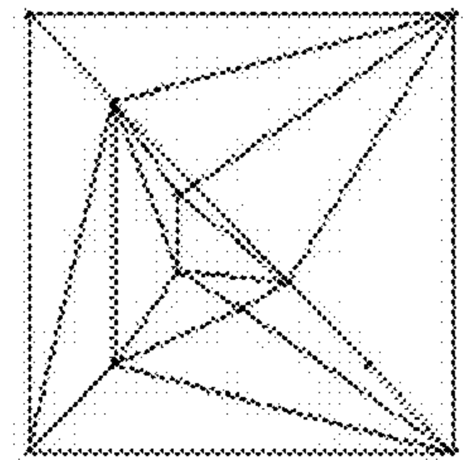


FIG. 12G

('961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12F and 12G is that a polygon (not a triangle) has been split.

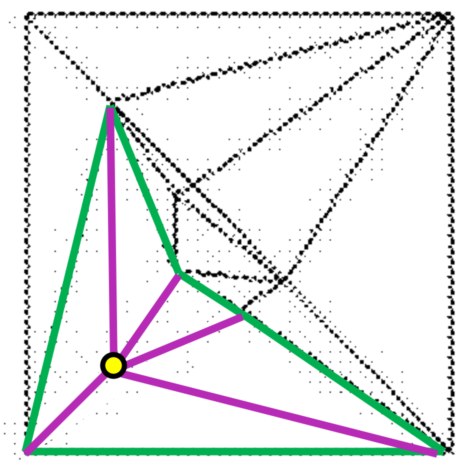


FIG. 12F Prior Art

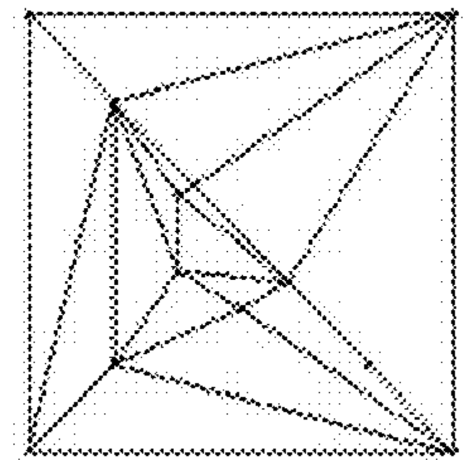


FIG. 12G

('961 Patent at Fig. 12.)

“modified Watson method”

The difference between Figures 12F and 12G is that a polygon (not a triangle) has been split.

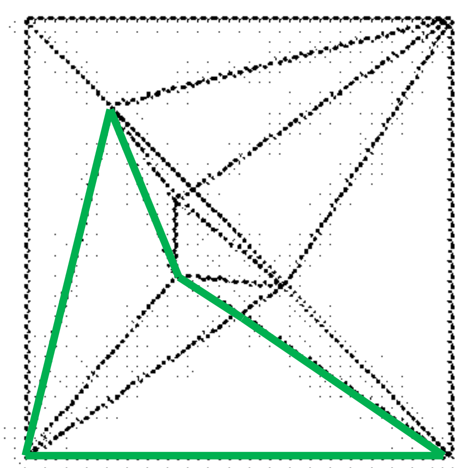


FIG. 12F Prior Art

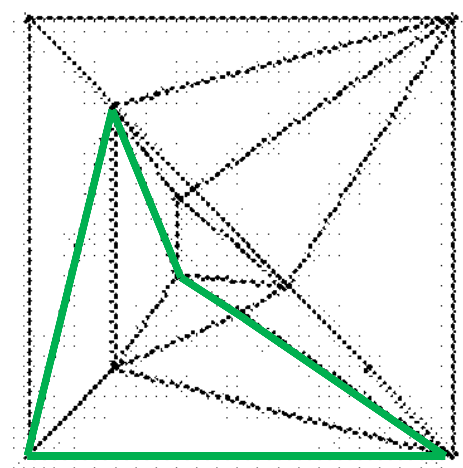


FIG. 12G

(’961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12G and 12H is that a **polygon** (not a triangle) has been split.

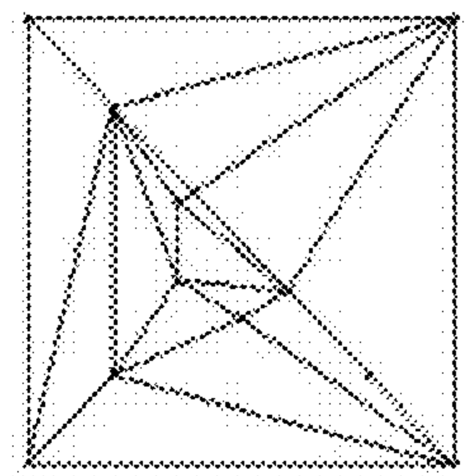


FIG. 12G

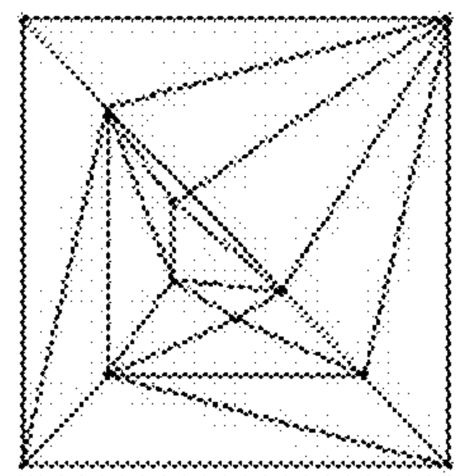


FIG. 12H

('961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12G and 12H is that a polygon (not a triangle) has been split.

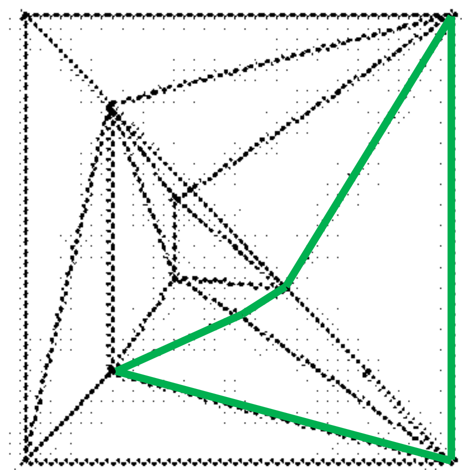


FIG. 12G

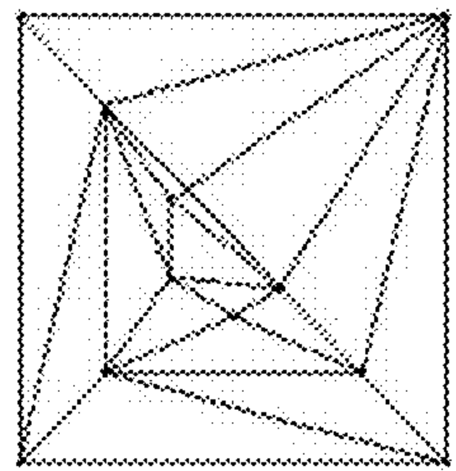


FIG. 12H

('961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12G and 12H is that a polygon (not a triangle) has been split.

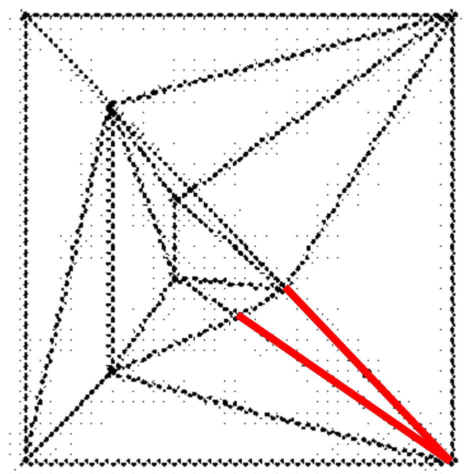


FIG. 12G

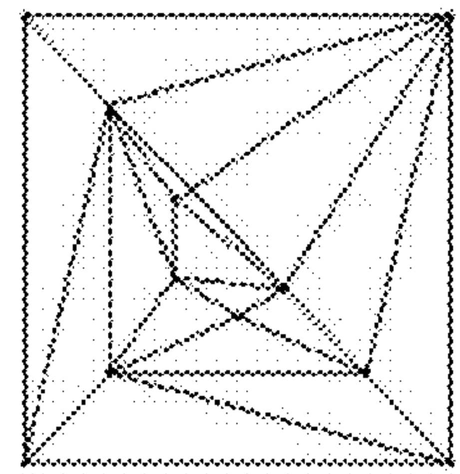


FIG. 12H

('961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12G and 12H is that a polygon (not a triangle) has been split.

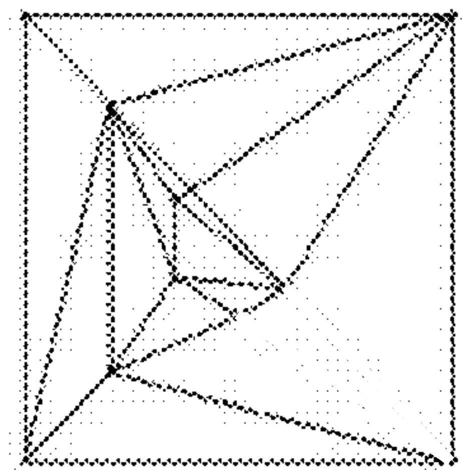


FIG. 12G

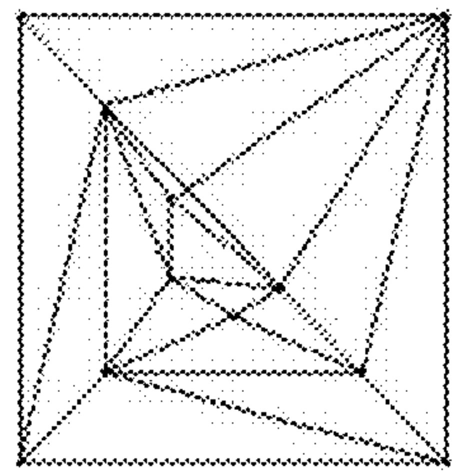


FIG. 12H

('961 Patent at Fig. 12.)

“modified Watson method”

The difference between Figures 12G and 12H is that a polygon (not a triangle) has been split.

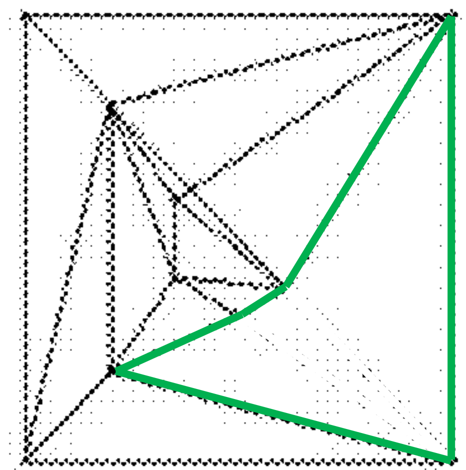


FIG. 12G

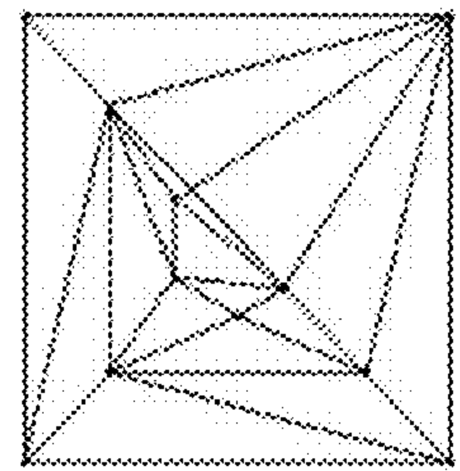


FIG. 12H

(’961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12G and 12H is that a polygon (not a triangle) has been split.

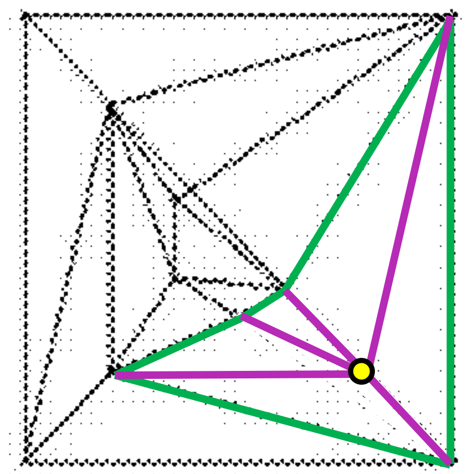


FIG. 12G

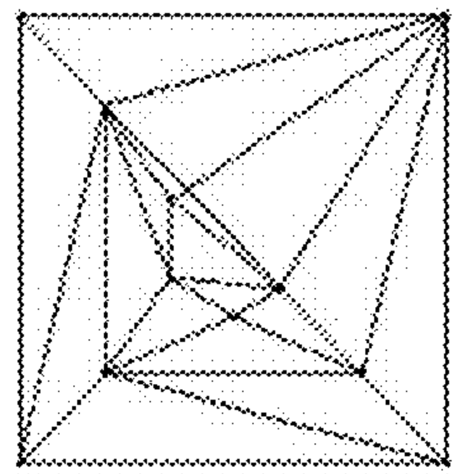


FIG. 12H

('961 Patent at Fig. 12.)

"modified Watson method"

The difference between Figures 12G and 12H is that a polygon (not a triangle) has been **split**.

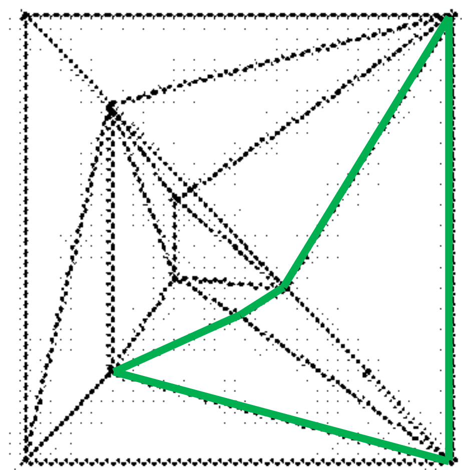


FIG. 12G

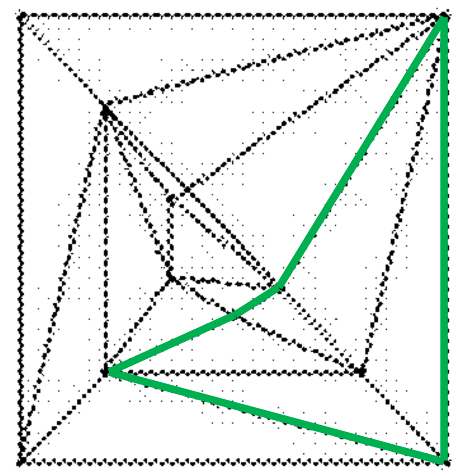


FIG. 12H

('961 Patent at Fig. 12.)

ADRMOP,AO279,APPEAL,CLOSED,CONSENT,ENETERM,PROTO

U.S. District Court
California Northern District (San Francisco)
CIVIL DOCKET FOR CASE #: 3:19-cv-03192-SK

Nature Simulation Systems Inc. v. Autodesk, Inc.
Assigned to: Magistrate Judge Sallie Kim
Case in other court: USCA Fed Cir, 20-02257
Cause: 35:271 Patent Infringement

Date Filed: 06/07/2019
Date Terminated: 08/11/2020
Jury Demand: Defendant
Nature of Suit: 830 Patent
Jurisdiction: Federal Question

Plaintiff

Nature Simulation Systems Inc. represented by

Todd Christopher Atkins
Atkins &Davidson, APC
2261 Rutherford Road
Carlsbad, CA 92008
(619) 665-3476
Email: tatkins@atkinsdavidson.com
LEAD ATTORNEY
ATTORNEY TO BE NOTICED

Matthew M. Wawrzyn
Wawrzyn & Jarvis LLC
2700 Patriot Boulevard, Suite 250
Glenview, IL 60026
847-656-5848
Email: matt@wawrzynlaw.com
PRO HAC VICE
ATTORNEY TO BE NOTICED

Stephen C Jarvis
Wawrzyn & Jarvis LLC
2700 Patriot Boulevard
Suite 250
Glenview, IL 60026
(847) 656-5864
Email: stephen@wawrzynlaw.com
TERMINATED: 10/01/2019
PRO HAC VICE

Stephen C. Jarvis
Wawrzyn & Jarvis LLC
2700 Patriot Boulevard, Suite 250
Glenview, IL 60026
847-656-5848
Email: stephen@wawrzynlaw.com
TERMINATED: 10/01/2019

V.

Defendant

Autodesk, Inc. represented by **Rudolph Kim**
Morrison & Foerster LLP
755 Page Mill Road
Palo Alto, CA 94304
(650) 813-5600
Fax: (650) 494-0792
Email: rudykim@mofo.com
LEAD ATTORNEY
ATTORNEY TO BE NOTICED

Roman A. Swoopes
Morrison & Foerster LLP
755 Page Mill Road
Palo Alto, CA 94304-1018
(650) 813-5600
Fax: (650) 494-0792
Email: rswoopes@mofo.com
ATTORNEY TO BE NOTICED

Stephen J. H. Liu
Morrison Foerster LLP
755 Page Mill Road
Palo Alto, CA 94304-1018
650-813-6493
Fax: 650-494-0792
Email: sliu@mofo.com
ATTORNEY TO BE NOTICED

Date Filed	#	Docket Text
06/07/2019	1	COMPLAINT FOR PATENT INFRINGEMENT against Defendant Autodesk, Inc., (Filing Fee: $400.00, receipt number 0971- 13418619). Filed by Nature Simulation Systems Inc.. (Attachments: #(1) Civil Cover Sheet)(Atkins, Todd) (Filed on 6/7/2019) Modified on 6/11/2019 (tnS, COURT STAFF). (Entered: 06/07/2019)
06/07/2019	2	Proposed Summons. (Atkins, Todd) (Filed on 6/7/2019) (Entered: 06/07/2019)
06/07/2019	3	Certification of Interested Entities or Persons Pursuant to Civil L.R. 3-15 filed by Nature Simulation Systems Inc.. (Atkins, Todd) (Filed on 6/7/2019) (Entered: 06/07/2019)
06/07/2019	4	Disclosure Statement Pursuant to Fed. R.Civ.P. 7.1 filed by Nature Simulation Systems Inc.. (Atkins, Todd) (Filed on 6/7/2019) (Entered: 06/07/2019)
06/07/2019	5	REPORT on the filing of an action regarding *Patent Infringement* (cc: form mailed to register). (Atkins, Todd) (Filed on 6/7/2019) (Entered: 06/07/2019)
06/07/2019	6	Case assigned to Magistrate Judge Sallie Kim. Counsel for plaintiff or the removing party is responsible for serving the Complaint or Notice of Removal, Summons and the assigned judge's standing orders and all other new case documents upon the opposing parties. For information, visit *E-Filing A New Civil Case* at http://cand.uscourts.gov/ecf/caseopening. Standing orders can be downloaded from the court's web page at www.cand.uscourts.gov/judges. Upon receipt, the summons will be issued and returned

		electronically. Counsel is required to send chambers a copy of the initiating documents pursuant to L.R. 5-1(e)(7). A scheduling order will be sent by Notice of Electronic Filing (NEF) within two business days. Consent/Declination due by 6/21/2019. (ajsS, COURT STAFF) (Filed on 6/7/2019) (Entered: 06/07/2019)
06/11/2019	7	**Initial Case Management Scheduling Order with ADR Deadlines: Joint Case Management Statement due by 9/2/2019. Initial Case Management Conference set for 9/9/2019 at 1:30 PM in San Francisco, Courtroom C, 15th Floor. (tnS, COURT STAFF) (Filed on 6/11/2019) (Entered: 06/11/2019)**
06/11/2019	8	Summons Issued as to Defendant Autodesk, Inc.. (tnS, COURT STAFF) (Filed on 6/11/2019) (Entered: 06/11/2019)
07/12/2019	9	CLERK'S NOTICE Re: Consent or Declination: Plaintiff shall file a consent or declination to proceed before a magistrate judge by 7/26/2019. Note that any party is free to withhold consent to proceed before a magistrate judge without adverse substantive consequences. The forms are available at: http://cand.uscourts.gov/civilforms. (mklS, COURT STAFF) (Filed on 7/12/2019) (Entered: 07/12/2019)
07/24/2019	10	STIPULATION *to Extend Time to Respond to Complaint* filed by Autodesk, Inc.. (Kim, Rudolph) (Filed on 7/24/2019) (Entered: 07/24/2019)
07/26/2019	11	CONSENT/DECLINATION to Proceed Before a US Magistrate Judge by Nature Simulation Systems Inc... (Atkins, Todd) (Filed on 7/26/2019) (Entered: 07/26/2019)
07/26/2019	12	CLERK'S NOTICE Re: Consent or Declination: Defendant shall file a consent or declination to proceed before a magistrate judge by 8/9/2019. Note that any party is free to withhold consent to proceed before a magistrate judge without adverse substantive consequences. The forms are available at: http://cand.uscourts.gov/civilforms. (mklS, COURT STAFF) (Filed on 7/26/2019) (Entered: 07/26/2019)
07/26/2019	13	CONSENT/DECLINATION to Proceed Before a US Magistrate Judge by Autodesk, Inc... (Kim, Rudolph) (Filed on 7/26/2019) (Entered: 07/26/2019)
08/19/2019	14	ADR Certification (ADR L.R. 3-5 b) of discussion of ADR options *filed on behalf of Autodesk, Inc*. (Kim, Rudolph) (Filed on 8/19/2019) (Entered: 08/19/2019)
08/20/2019	15	ADR Certification (ADR L.R. 3-5 b) of discussion of ADR options (Atkins, Todd) (Filed on 8/20/2019) (Entered: 08/20/2019)
08/30/2019	16	NOTICE of Appearance by Roman A Swoopes *for defendant Autodesk, Inc*. (Swoopes, Roman) (Filed on 8/30/2019) (Entered: 08/30/2019)
08/30/2019	17	ANSWER to 1 Complaint with Jury Demand; by Autodesk, Inc. (Kim, Rudolph) (Filed on 8/30/2019) Modified on 9/3/2019 (aaaS, COURT STAFF). (Entered: 08/30/2019)
08/30/2019	18	Certificate of Interested Entities by Autodesk, Inc. (Kim, Rudolph) (Filed on 8/30/2019) (Entered: 08/30/2019)
08/30/2019	19	Rule 7.1 Disclosures by Autodesk, Inc. (Kim, Rudolph) (Filed on 8/30/2019) (Entered: 08/30/2019)
08/30/2019		Electronic filing error. Corrected by Clerk's Office. No further action is necessary. Re: 16 Notice of Appearance filed by Nature Simulation Systems Inc. **Attorney Add ed Roman A. Swoopes to represent Plaintiff Instead of Defendant.** (aaaS, COURT STAFF) (Filed on 8/30/2019) (Entered: 09/03/2019)
09/03/2019	20	JOINT CASE MANAGEMENT STATEMENT filed by Nature Simulation Systems Inc.. (Atkins, Todd) (Filed on 9/3/2019) (Entered: 09/03/2019)

09/03/2019	21	MOTION & [PROPOSED] ORDER for leave to appear in Pro Hac Vice (Filing fee $ 310, receipt number 0971-13669923.) filed by Nature Simulation Systems Inc.. (Attachments: # 1 Certificate of Good Standing)(Jarvis, Stephen) (Filed on 9/3/2019) Modified on 9/4/2019 (aaaS, COURT STAFF). (Entered: 09/03/2019)
09/03/2019	22	**ORDER by Magistrate Judge Sallie Kim granting 21 Motion for Pro Hac Vice. (sklc2S, COURT STAFF) (Filed on 9/3/2019) (Entered: 09/03/2019)**
09/09/2019	23	**Minute Entry for proceedings held before Magistrate Judge Sallie Kim: Initial Case Management Conference held on 9/9/2019. Case Management Statement due by 5/26/2020. Further Case Management Conference set for 6/1/2020 01:30 PM in San Francisco, Courtroom C, 15th Floor. Claims Construction Hearing set for 4/23/2020 10:00 AM. Tutorial Hearing set for 4/9/2020 10:00 AM in San Francisco, Courtroom C, 15th Floor.FTR Time: 1:59 - 2:02. Plaintiff Attorney: Stephen Jarvis. Defendant Attorney: Rudolph Kim and Roman Swoopes. Attachment: Civil Minutes. (tlS, COURT STAFF) (Date Filed: 9/9/2019) (Entered: 09/10/2019)**
09/09/2019	24	**ORDER SETTING SCHEDULE re 23 Initial Case Management Conference. Signed by Judge Sallie Kim on 9/9/2019. (tlS, COURT STAFF) (Filed on 9/9/2019) (Entered: 09/10/2019)**
10/01/2019	25	NOTICE of Change In Counsel by Todd Christopher Atkins (Atkins, Todd) (Filed on 10/1/2019) (Entered: 10/01/2019)
01/06/2020	27	STIPULATION AND [PROPOSED] ORDER *to Modify Claim Construction Schedule* filed by Autodesk, Inc.. (Kim, Rudolph) (Filed on 1/6/2020) (Entered: 01/06/2020)
01/07/2020	28	**ORDER by Judge Sallie Kim granting 27 Stipulation to Modify Claim Construction Schedule.Tutorial Hearing set for 5/26/2020 10:00 AM. Claims Construction Hearing set for 6/4/2020 10:00 AM. Further Case Management Conference set for 7/13/2020 01:30 PM. All hearings in San Francisco, Courtroom C, 15th Floor. Case Management Statement due by 7/6/2020. (sklc2S, COURT STAFF) (Filed on 1/7/2020) (Entered: 01/07/2020)**
03/12/2020	29	ADR Clerk's Notice Appointing Shella Deen as ENE Evaluator. (af, COURT STAFF) (Filed on 3/12/2020) (Entered: 03/12/2020)
03/12/2020	30	MOTION for Protective Order *STIPULATED [PROPOSED] PROTECTIVE ORDER* filed by Autodesk, Inc.. Responses due by 3/26/2020. Replies due by 4/2/2020. (Attachments: # 1 Declaration of Roman A. Swoopes ISO Stipulated Protective Order, # 2 Exhibit A to Declaration of Roman A. Swoopes ISO Stipulated Protective Order)(Kim, Rudolph) (Filed on 3/12/2020) (Entered: 03/12/2020)
03/13/2020	31	CLAIM CONSTRUCTION STATEMENT *Jointly Submitted* filed by Nature Simulation Systems Inc.. (Attachments: # 1 Exhibit Claim Construction Chart, # 2 Exhibit 961 Patent, # 3 Exhibit 105 Patent)(Atkins, Todd) (Filed on 3/13/2020) (Entered: 03/13/2020)
03/18/2020	32	**STIPULATED PROTECTIVE ORDER (granting 30 Stipulation) by Magistrate Judge Sallie Kim. (mklS, COURT STAFF) (Filed on 3/18/2020) (Entered: 03/18/2020)**
04/06/2020	33	STIPULATION WITH PROPOSED ORDER *to Extend Claim Construction Discovery Deadline* filed by Nature Simulation Systems Inc.. (Atkins, Todd) (Filed on 4/6/2020) (Entered: 04/06/2020)
04/06/2020	34	**ORDER by Magistrate Judge Sallie Kim granting 33 Stipulation to Extend Claim Construction Discovery Deadline. (mklS, COURT STAFF) (Filed on 4/6/2020) (Entered: 04/06/2020)**
04/07/2020		Set/Reset Hearing ENE Hearing set for 7/21/2020 09:30 AM., at the law offices of Hoge,

		Fenton, Jones & Appel, Inc., 60 South Market Street, Suite 1400, San Jose, CA 95113. (af, COURT STAFF) (Filed on 4/7/2020) (Entered: 04/07/2020)
04/27/2020	35	CLAIM CONSTRUCTION STATEMENT *Opening Brief* filed by Nature Simulation Systems Inc.. (Atkins, Todd) (Filed on 4/27/2020) (Entered: 04/27/2020)
05/11/2020	36	CLAIM CONSTRUCTION STATEMENT - *Autodesk, Inc.'s Responsive Claim Construction Brief* filed by Autodesk, Inc.. (Attachments: # 1 Declaration of Daniel G. Aliaga, # 2 Exhibit A to the Aliaga Declaration)(Kim, Rudolph) (Filed on 5/11/2020) (Entered: 05/11/2020)
05/18/2020	37	CLAIM CONSTRUCTION STATEMENT *Reply Claim Construction Brief* filed by Nature Simulation Systems Inc.. (Attachments: # 1 Exhibit Exhibit A)(Atkins, Todd) (Filed on 5/18/2020) (Entered: 05/18/2020)
05/18/2020	38	CLERKS NOTICE SETTING ZOOM HEARING: Notice is given that the Tutorial on 5/26/2020 at 10:00 AM will be a Zoom video conferencing webinar. ADVANCE REGISTRATION OF PARTICIPATING COUNSEL IS REQUIRED and must be done by emailing a list of names and emails of counsel who will be participating in the hearing to SKCRD@cand.uscourts.gov by no later than Thursday May 21, 2020 at 3:00 PM Pacific. Case participants may also receive an email invitation from the court with different information which should be followed. All counsel, members of the public and press please click the link or use the information below to join the webinar: https://us02web.zoom.us/j/85216500668? pwd=SDZxL1dEQTVZUTRWbFFTZWNDZWFwdz09 Meeting ID: 837 6542 3592 Password: 253885 Dial by your location +1 929 205 6099 US (New York) +1 253 215 8782 US +1 301 715 8592 US +1 312 626 6799 US (Chicago) +1 346 248 7799 US (Houston) +1 669 900 6833 US (San Jose) Find your local number: https://zoom.us/u/ac4JkPfcjo For Zoom connection, see: https://apps.cand.uscourts.gov/telhrg/ For important information and guidance on technical preparation, please see https://www.cand.uscourts.gov/zoom/. PLEASE NOTE: Persons granted remote access to court proceedings are reminded of the general prohibition against photographing, recording, and rebroadcasting of court proceedings (including those held by telephone or videoconference). See General Order 58 at Paragraph III. Any recording of a court proceeding held by video or teleconference, including screen-shots or other visual copying of a hearing, is absolutely prohibited. Violation of these prohibitions may result in sanctions, including removal of court-issued

		media credentials, restricted entry to future hearings, or any other sanctions deemed necessary by the court. *(This is a text-only entry generated by the court. There is no document associated with this entry.)* (mklS, COURT STAFF) (Filed on 5/18/2020) (Entered: 05/18/2020)
05/19/2020	39	MOTION for leave to appear in Pro Hac Vice (Filing fee $ 310, receipt number 0971-14485623.) filed by Nature Simulation Systems Inc.. (Atkins, Todd) (Filed on 5/19/2020) (Entered: 05/19/2020)
05/19/2020	40	****Filed in error without judge's signature. Refer to docket no. 41 .** ORDER by Magistrate Judge Sallie Kim granting 39 Motion for Pro Hac Vice. (sklc2S, COURT STAFF) (Filed on 5/19/2020) Modified on 5/20/2020 (mklS, COURT STAFF). Modified on 5/20/2020 (mklS, COURT STAFF). (Entered: 05/19/2020)**
05/20/2020	41	**ORDER (with signature affixed) by Magistrate Judge Sallie Kim granting 39 Motion for Pro Hac Vice as to Matthew Wawrzyn. (mklS, COURT STAFF) (Filed on 5/20/2020) (Entered: 05/20/2020)**
05/26/2020	42	CLERK'S NOTICE Re: Tutorial: The Court requests that both sides file the slideshows presented at today's tutorial on the docket. *(This is a text-only entry generated by the court. There is no document associated with this entry.)* (mklS, COURT STAFF) (Filed on 5/26/2020) (Entered: 05/26/2020)
05/26/2020	43	**Minute Entry for proceedings held before Magistrate Judge Sallie Kim: Tutorial Hearing by Zoom held on 5/26/2020. Claim Construction Hearing by previously set for 6/4/2020 at 10:00 AM and will be held by Zoom. (Total Time in Court: 1 hour, 5 minutes.) (Court Reporter: Belle Ball.) Attorney for Plaintiff: Matthew Wawrzyn. Attorneys for Defendant: Rudy Kim; Roman Swoopes. Also Present: Dr. Daniel Aliaga (Defense Expert). *(This is a text-only entry generated by the court. There is no document associated with this entry.)* (mklS, COURT STAFF) (Date Filed: 5/26/2020) (Entered: 05/26/2020)**
05/26/2020	44	CLAIM CONSTRUCTION STATEMENT*Tutorial Presentation by Shangwen Cao* filed by Nature Simulation Systems Inc.. (Atkins, Todd) (Filed on 5/26/2020) (Entered: 05/26/2020)
05/26/2020	45	CLAIM CONSTRUCTION STATEMENT --*Autodesk's Technical Tutorial slides* filed by Autodesk, Inc.. (Kim, Rudolph) (Filed on 5/26/2020) (Entered: 05/26/2020)
05/26/2020	46	NOTICE of Appearance by Stephen J. H. Liu *for Autodesk, Inc*. (Liu, Stephen) (Filed on 5/26/2020) (Entered: 05/26/2020)
05/28/2020	47	CLERKS NOTICE SETTING ZOOM HEARING: Notice is given that the Claims Construction Hearing set for 6/4/2020 10:00 AM will be a Zoom video conferencing webinar. Case participants may also receive an email invitation from the court with different information which should be followed. All counsel, members of the public and press please click the link or use the information below to join the webinar: https://us02web.zoom.us/j/81695787620?pwd=UkJnY0w5RnF6ZEdiSUszSHQ3MkI1Zz09

		Meeting ID: 816 9578 7620 Password: 253885 Dial by your location +1 929 205 6099 US (New York) +1 253 215 8782 US +1 301 715 8592 US +1 312 626 6799 US (Chicago) +1 346 248 7799 US (Houston) +1 669 900 6833 US (San Jose) Find your local number: https://zoom.us/u/ac4JkPfcjo For important information and guidance on technical preparation, please see https://www.cand.uscourts.gov/zoom/. For Zoom c onnection, see: https://apps.cand.uscourts.gov/telhrg/ PLEASE NOTE: Persons granted remote access to court proceedings are reminded of the general prohibition against photographing, recording, and rebroadcasting of court proceedings (including those held by telephone or videoconference). See General Order 58 at Paragraph III. Any recording of a court proceeding held by video or teleconference, including screen-shots or other visual copying of a hearing, is absolutely prohibited. Violation of these prohibitions may result in sanctions, including removal of court-issued media credentials, restricted entry to future hearings, or any other sanctions deemed necessary by the court. *(This is a text-only entry generated by the court. There is no document associated with this entry.)* (mklS, COURT STAFF) (Filed on 5/28/2020) (Entered: 05/28/2020)
06/03/2020	48	CLAIM CONSTRUCTION STATEMENT *Amended Joint Claim Construction and Prehearing Statement* filed by Nature Simulation Systems Inc.. (Attachments: # 1 Exhibit Amended Attachment A)(Atkins, Todd) (Filed on 6/3/2020) (Entered: 06/03/2020)
06/04/2020	49	**Minute Entry for proceedings held before Magistrate Judge Sallie Kim: Claims Construction / Markman Hearing by Zoom held on 6/4/2020. The Court took the first two terms under submission and will issue a written order. A further Claims Construction hearing will be set regarding the remaining terms, if needed. The Court requests that the parties file their slide presentation on the docket. (Total Time in Court: 2 hours, 24 minutes.) (Court Reporter: Ruth Levine Ekhaus.)** **Attorney for plaintiff: Matthew Wawrzyn.** **Attorney for Defendant: Rudy Kim; Roman Swoopes.** ***(This is a text-only entry generated by the court. There is no document associated with this entry.)* (mklS, COURT STAFF) (Date Filed: 6/4/2020) (Entered: 06/04/2020)**
06/04/2020	50	CLAIM CONSTRUCTION STATEMENT *Claim Construction Presentation by Nature Simulation Systems Inc.* filed by Nature Simulation Systems Inc.. (Atkins, Todd) (Filed on 6/4/2020) (Entered: 06/04/2020)
06/04/2020	51	CLAIM CONSTRUCTION STATEMENT --*Autodesk, Inc.'s Claim Construction Presentation* filed by Autodesk, Inc.. (Kim, Rudolph) (Filed on 6/4/2020) (Entered: 06/04/2020)

06/05/2020	52	NOTICE by Autodesk, Inc. - *Autodesk's Notice of Supplemental Authority* (Attachments: # 1 Attachment A, # 2 Attachment B, # 3 Attachment C)(Kim, Rudolph) (Filed on 6/5/2020) (Entered: 06/05/2020)
06/24/2020	53	TRANSCRIPT ORDER for proceedings held on 6/4/2020 before Magistrate Judge Sallie Kim by Autodesk, Inc., for Court Reporter Ruth Ekhaus. (Swoopes, Roman) (Filed on 6/24/2020) (Entered: 06/24/2020)
07/06/2020	54	JOINT CASE MANAGEMENT STATEMENT filed by Nature Simulation Systems Inc.. (Atkins, Todd) (Filed on 7/6/2020) (Entered: 07/06/2020)
07/07/2020	55	STIPULATION WITH PROPOSED ORDER *to Extend Deadline to Complete Early Neutral Evaluation* filed by Autodesk, Inc.. (Kim, Rudolph) (Filed on 7/7/2020) (Entered: 07/07/2020)
07/08/2020	56	CLERKS NOTICE SETTING ZOOM HEARING: Notice is hereby given that the Further Case Management Conference set for 7/13/2020 01:30 PM before Magistrate Judge Sallie Kim will be a Zoom video conferencing webinar. For Zoom connection, see: https://apps.cand.uscourts.gov/telhrg/ PLEASE NOTE: Persons granted access to court proceedings held by telephone or videoconference are reminded that photographing, recording, and rebroadcasting of court proceedings, including screenshots or other visual copying of a hearing, is absolutely prohibited. See General Order 58 at Paragraph III. All counsel, members of the public and press please click the link or use the information below to join the webinar: https://cand-uscourts.zoomgov.com/j/1606612971? pwd=U0o4Z3FmSU4zcmE1MjMzMjN5dllEdz09 Meeting ID: 160 661 2971 Password: 028445 Dial by your location +1 929 205 6099 US (New York) +1 253 215 8782 US +1 301 715 8592 US +1 312 626 6799 US (Chicago) +1 346 248 7799 US (Houston) +1 669 900 6833 US (San Jose) Find your local number: https://zoom.us/u/ac4JkPfcjo For important information and guidance on technical preparation, please see https://www.cand.uscourts.gov/zoom/. *(This is a text-only entry generated by the court. There is no document associated with this entry.)* (mklS, COURT STAFF) (Filed on 7/8/2020) (Entered: 07/08/2020)
07/13/2020	57	**Minute Entry for proceedings held before Magistrate Judge Sallie Kim: Further Case Management Conference held on 7/13/2020. The Court advised the parties that it is going to find that the claim terms are indefinite. The Court will issue a written order by the end of July. Further Case Management Conference set for 8/3/2020 01:30 PM. Updated joint case management statement due by 7/27/2020. By that time the parties will have received the orde** [illegible] **ered how they would like to proceed, including**

		whether a settlement conference with a magistrate judge would be helpful. (Zoom Recording Time: 1:31-1:37.) **Attorney for Plaintiff: Matthew Wawrzyn.** **Attorneys for Defendant: Rudy Kim; Roman Swoopes.** **For Zoom connection, see: https://apps.cand.uscourts.gov/telhrg/** **This proceeding will be a Zoom video conferencing webinar.** **PLEASE NOTE: Persons granted access to court proceedings held by telephone or videoconference are reminded that photographing, recording, and rebroadcasting of court proceedings, including screenshots or other visual copying of a hearing, is absolutely prohibited. See General Order 58 at Paragraph III.** **All counsel, members of the public and press please click the link or use the information below to join the webinar:** **https://cand-uscourts.zoomgov.com/j/1606612971?pwd=U0o4Z3FmSU4zcmE1MjMzMjN5dllEdz09** **Meeting ID: 160 661 2971** **Password: 028445** **Dial by your location** **+1 929 205 6099 US (New York)** **+1 253 215 8782 US** **+1 301 715 8592 US** **+1 312 626 6799 US (Chicago)** **+1 346 248 7799 US (Houston)** **+1 669 900 6833 US (San Jose)** **Find your local number: https://zoom.us/u/ac4JkPfcjo** **For important information and guidance on technical preparation, please see https://www.cand.uscourts.gov/zoom/.** ***(This is a text-only entry generated by the court. There is no document associated with this entry.)* (mklS, COUR T STAFF) (Date Filed: 7/13/2020) (Entered: 07/13/2020)**
07/14/2020	58	**ORDER by Magistrate Judge Sallie Kim granting 55 Stipulation and Extending ADR Deadline. (sklc2S, COURT STAFF) (Filed on 7/14/2020) (Entered: 07/14/2020)**
07/15/2020	59	ADR Remark: The deadline to complete the ENE was extended to 11/30/2020 58 . The ENE scheduled on 7/21/2020 is OFF CALENDAR, and will be rescheduled. (af, COURT STAFF) (Filed on 7/15/2020) *(This is a text-only entry generated by the court. There is no document associated with this entry.)* (Entered: 07/15/2020)
07/27/2020	60	JOINT CASE MANAGEMENT STATEMENT filed by Nature Simulation Systems Inc.. (Atkins, Todd) (Filed on 7/27/2020) (Entered: 07/27/2020)
07/31/2020	61	**CLAIM CONSTRUCTION ORDER. Signed by Magistrate Judge Sallie Kim on 7/31/20. (sklc2S, COURT STAFF) (Filed on 7/31/2020) (Entered: 07/31/2020)**
07/31/2020	62	Transcript of Proceedings held on 06/04/2020, before Judge Sallie Kim. Court Reporter Ruth Levine Ekhaus, RDR, FCRR, CSR No. 12219, telephone number (415)336-5223/ ruth_ekhaus@can r General Order No. 59 and Judicial Conference

		policy, this transcript may be viewed only at the Clerk's Office public terminal or may be purchased through the Court Reporter until the deadline for the Release of Transcript Restriction. After that date, it may be obtained through PACER. Any Notice of Intent to Request Redaction, if required, is due no later than 5 business days from date of this filing. (Re 53 Transcript Order) Release of Transcript Restriction set for 10/29/2020. (Related documents(s) 53) (rreS, COURT STAFF) (Filed on 7/31/2020) (Entered: 07/31/2020)
08/03/2020	63	**Minute Entry for proceedings held before Magistrate Judge Sallie Kim: Further Case Management Conference by Zoom held on 8/3/2020. Plaintiff anticipates appealing the claim construction order. Parties should submit a proposed stipulated judgment for the Court's consideration. (Zoom Recording Time: 1:35-1:41.)** **Attorney for Plaintiff: Matthew Wawrzyn.** **Attorney for Defendant: Rudy Kim; Roman Swoopes.** ***(This is a text-only entry generated by the court. There is no document associated with this entry.)*** **(mklS, COURT STAFF) (Date Filed: 8/3/2020) (Entered: 08/03/2020)**
08/10/2020	64	STIPULATION WITH PROPOSED ORDER *to Extend the Deadlines for Proceedings Regarding Attorneys' Fees and Costs* filed by Autodesk, Inc.. (Attachments: # 1 Proposed Judgment)(Kim, Rudolph) (Filed on 8/10/2020) (Entered: 08/10/2020)
08/11/2020	65	**ORDER by Magistrate Judge Sallie Kim granting 64 Stipulation to Extend the Deadlines for Proceedings Regarding Attorneys' Fees and Costs. (mklS, COURT STAFF) (Filed on 8/11/2020) (Entered: 08/11/2020)**
08/11/2020	66	**JUDGMENT. Signed by Magistrate Judge Sallie Kim on 8/11/2020. (mklS, COURT STAFF) (Filed on 8/11/2020) (Entered: 08/11/2020)**
08/11/2020	67	ADR Clerk's Notice Vacating Appointment of Evaluator Shella Deen (af, COURT STAFF) (Filed on 8/11/2020) (Entered: 08/11/2020)
08/12/2020	68	REPORT on the determination of an action regarding patent infringement (cc: form mailed to register). (Attachments: # 1 Order, # 2 Judgment)(slhS, COURT STAFF) (Filed on 8/12/2020) (Entered: 08/12/2020)
09/04/2020	69	NOTICE OF APPEAL to the Federal Circuit by Nature Simulation Systems Inc.. Filing fee $ 505, receipt number 0971-14903341. Appeal of Order 65 , Judgment 66 (Atkins, Todd) (Filed on 9/4/2020) Modified on 9/8/2020 (slhS, COURT STAFF). **(USCA Case No. 2020-2257)** (Entered: 09/04/2020)
09/08/2020	70	Notice of Appeal and Docket Sheet transmitted to US Court of Appeals re 69 Notice of Appeal to the Federal Circuit *(This is a text-only entry generated by the court. There is no document associated with this entry.)* (slhS, COURT STAFF) (Filed on 9/8/2020) (Entered: 09/08/2020)
09/15/2020	71	USCA Case Number 2020-2257 for 69 Notice of Appeal to the Federal Circuit filed by Nature Simulation Systems Inc. (slhS, COURT STAFF) (Filed on 9/15/2020) (Entered: 09/15/2020)

PACER Service Center			
Transaction Receipt			
03/09/2021 12:32:36			
PACER Login:	mwawrzyn:3585925:0	**Client Code:**	