

(51) International Patent Classification:  
G06T 17/20 (2006.01)(21) International Application Number:  
PCT/US2020/012422(22) International Filing Date:  
06 January 2020 (06.01.2020)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicants: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 10300 Energy Drive, Spring, Texas 77389 (US). **UNIVERSIDADE FEDERAL DE PERNAMBUCO** [BR/BR]; Ave. Prof. Moraes Rego, 1235-Didade Universitaria, 50670-901 Recife (BR).(72) Inventors: **MELO, Silvio B.**; Av. Jorn. Anibal Fernandes, 50740-560 Pernambuco (BR). **CHARAMBA, Luiz G.R.**; Av. Jorn. Anibal Fernandes, 50740-560 Pernambuco (BR). **HENRIQUES, Diogo B.**; Av. Jorn. Anibal Fernandes, 50740-560 Pernambuco (BR). **WHITE, Scott A.**; 11311 W. Chinden Blvd., Boise, Idaho 83714 (US).(74) Agent: **WOODWORTH, Jeffrey C.** et al.; Mail Stop 35, 3390 E. Harmony Road, Fort Collins, Colorado 80528 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ,

CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

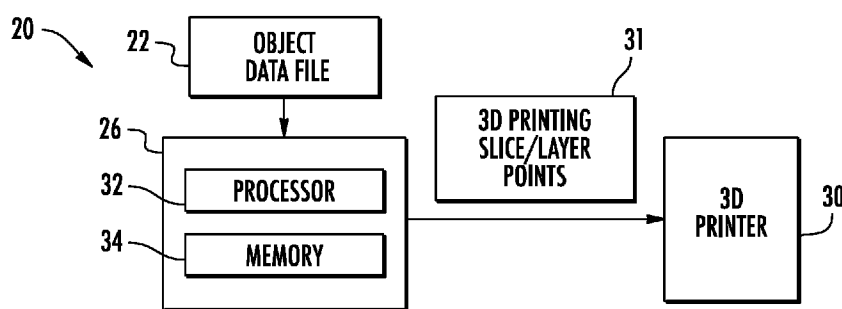
**Declarations under Rule 4.17:**

— as to the identity of the inventor (Rule 4.17(i))

**Published:**

— with international search report (Art. 21(3))

(54) Title: GENERATING 3D PRINTING POINTS

**FIG. 1**

(57) **Abstract:** A method for generating 3D printing points may include obtaining a Steiner patch that is part of a tessellation approximation of the 3D object, determining a parametric curve of a slicing plane and the Steiner patch, determining a classification of the parametric curve, sampling, based upon the classification, first and second points spaced by a parametric spacing along the parametric curve, determining a Euclidean spacing of the first and second points, and comparing the Euclidean spacing to a predefined spacing threshold. In response to the Euclidean spacing failing to satisfy the predefined threshold, sampling a third point along the parametric curve between the first and second points, generating 3D printing points in Euclidean space for the object based upon the first point, second point and third point sampled along the parametric curve.



## GENERATING 3D PRINTING POINTS

### BACKGROUND

[0001] Three-dimensional (3D) printing is sometimes used to print a three-dimensional object on a layer-by-layer basis. Each layer corresponds to a slice of the three-dimensional object being printed. Each slice comprises printing points that are generated from a digital file of the three-dimensional object being printed. The generated printing points are used by a three-dimensional printer to print the three-dimensional object.

### BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Figure 1 is a block diagram schematically illustrating portions of an example 3D printing system.

[0003] Figure 2 is a flow diagram of an example 3D printing point generation method.

[0004] Figure 3A is a perspective view of an example sphere approximated by a mesh of Steiner patches.

[0005] Figure 3B is a perspective view of an individual Steiner patch of the example sphere of Figure 3A, together with its control points and edges.

[0006] Figure 4A is a diagram of an example Steiner patch intersected by a slicing plane along a slicing curve in Euclidean space.

[0007] Figure 4B is a diagram of an example parametric space including a parametric curve corresponding to the slicing curve of Figure 4A.

[0008] Figure 5 is a diagram illustrating examples of conic classifications.

[0009] Figure 6 is a flow diagram of an example 3D printing point generation method.

[0010] Figure 7A is a diagram of an example parametric space including a parametric curve along which parametric points are being sampled.

[0011] Figure 7B is a diagram of an example Steiner patch being intersected by a slicing plane along a slicing curve, slicing curve corresponding to the parametric curve of Figure 7A, wherein the slicing curve includes Euclidean points corresponding to the parametric points of Figure 7A.

[0012] Figure 8A is a diagram of an example parametric space including a parametric curve along which parametric points are being sampled.

[0013] Figure 8B is a diagram of an example Steiner patch being intersected by a slicing plane along a slicing curve, slicing curve corresponding to the parametric curve of Figure 8A, wherein the slicing curve includes Euclidean points corresponding to the parametric points of Figure 8A.

[0014] Figure 9A is a diagram of an example parametric space including a parametric curve along which parametric points are being sampled.

[0015] Figure 9B is a diagram of an example Steiner patch being intersected by a slicing plane along a slicing curve, slicing curve corresponding to the parametric curve of Figure 9A, wherein the slicing curve includes Euclidean points corresponding to the parametric points of Figure 9A.

[00016] Figure 10A is a diagram of an example parametric space including a parametric curve along which parametric points are being sampled.

[00017] Figure 10B is a diagram of an example Steiner patch being intersected by a slicing plane along a slicing curve, slicing curve corresponding to the parametric curve of Figure 10A, wherein the slicing curve includes Euclidean points corresponding to the parametric points of Figure 10A.

[00018] Figure 11 is a flow diagram of an example 3D printing point generation method.

[00019] Figure 12 is a diagram illustrating reclassification of a parametric curve having an example non-degenerate conic classification to a degenerate conic classification.

[00020] Figure 13A is a diagram illustrating an example Euclidean space having an example Steiner patch intersected by a slicing plane along a slicing curve.

[00021] Figure 13B is a diagram of an example parametric space including an example parametric curve corresponding to the slicing curve of Figure 13A.

[00022] Throughout the drawings, identical reference numbers designate similar, but not necessarily identical, elements. The figures are not necessarily to scale, and the size of some parts may be exaggerated to more clearly illustrate the example shown. Moreover, the drawings provide examples and/or implementations consistent with the description; however, the description is not limited to the examples and/or implementations provided in the drawings.

## DETAILED DESCRIPTION OF EXAMPLES

[00023] Generating the printing points that form the slice of the three-dimensional object and that are used by the three-dimensional printer to print the three-dimensional object uses large amounts of memory, is often computationally burdensome and may be prone to inaccuracies. Generating such printing points necessitates use of a computational device or computer as such computations may be mathematically intense and may involve thousands to millions of different printing points for each slice of the three-dimensional object being printed.

[00024] An object may be modeled in a computer-assisted design (CAD) application. The CAD application may create a non-uniform rational basis spline (NURBS) representation of the object. A NURBS representation format may offer flexibility and precision in modeling the object. The NURBS representation may capture the surface area of the object being represented. A mesh format may be used to represent the same object. The mesh format may comprise multiple polygons connected along their sides. For example, different sizes and shapes of triangles may be used to represent the surface area of the object. The NURBS representation may be able to better capture curves of the object, while the mesh may approximate curves by using small polygons. The mesh may be more easily manipulated for other purposes, such as determining an intersection with a plane.

[00025] 3D objects to be printed are sometimes defined by a tessellation or mesh of Steiner patches. The mesh of Steiner patches is sometimes referred to as a non-uniform rational quadratic splines (NURQS) model. A Steiner Patch, also known as a rational quadratic Bézier triangle, may be used to model the object to allow for more closely matching curves than planar triangles, while maintaining the operation of intersection with a plane as compared to a NURBS representation. A planar triangle may be represented by three points in Euclidean space, one per corner of the planar triangle. A point may include three numbers for coordinates, an x, y, and z.

These coordinates may be represented by floating point numbers, or nine floating point numbers per planar triangle. A Steiner Patch may be represented by 21 floating point numbers, such as three corner points in Euclidean space, three intermediary control points in Euclidean space, and one weight per intermediary control point. As the Steiner Patch allows modeling of a 3D curved triangle, modeling an object using a mesh of Steiner Patches may use less storage than modeling an object using a mesh of planar triangles, while doing so with a higher accuracy. Other curved triangles may also be used, such as a rational cubic Bézier triangle, or other higher degrees of curved triangles. Although Steiner patches may offer enhanced accuracy (smoother edges) as compared to polygon meshes and may simplify the representation of the 3D object (simpler intersection calculation and less computational burden) as compared to NURBS, generation of 3D printing points from Steiner patches or NURBS models may still be computationally intensive and burdensome.

[00026] Disclosed are example 3D printing points determination instructions, systems and computer implemented methods that may generate printing points of 3D slices of an object, using Steiner patches, with less computational resources and with enhanced 3D print accuracy. The example instructions, systems and computer implemented methods reduce the computational burden while enhancing accuracy: (1) by limiting the sampling of points along a parametric curve of the intersecting plane and a Steiner patch in parametric space to points between border points of the parametric curve; (2) by iteratively adding parametric sampling points along the parametric curve in parametric space until the spacing between each pair of consecutive corresponding Euclidean points in Euclidean space satisfies a predefined threshold; and/or (3) reclassifying a non-degenerate conic parametric curve as a degenerate conic parametric curve for sampling the parametric points along the parametric curve.

[00027] Disclosed is an example non-transitory computer-readable medium providing instructions for a processor. The instructions direct the

processor to obtain a Steiner patch that is part of a tessellation approximation of a three-dimensional object to be printed by a 3D printer, determine a slicing curve, the slicing curve being an intersection of a slicing plane and the Steiner patch in Euclidian space, determine a parametric curve of the slicing curve, determine a classification of the parametric curve, determine border points of the parametric curve, sample points along the parametric curve between the border points of the parametric curve based on the classification, generate 3D printing points in Euclidean space for the object based upon the sampled points.

[00028] Disclosed is an example method for generating 3D printing points from Steiner patches. The example method may include obtaining a Steiner patch that is part of a tessellation approximation of the 3D object, determining a parametric curve of a slicing plane and the Steiner patch, determining a classification of the parametric curve, sampling, based upon the classification, first and second points spaced by a parametric spacing along the parametric curve, determining a Euclidean spacing of the first and second points, and comparing the Euclidean spacing to a predefined spacing threshold. In response to the Euclidean spacing failing to satisfy the predefined threshold, sampling a third point along the parametric curve between the first and second points, generating 3D printing points in Euclidean space for the object based upon the first point, second point and third point sampled along the parametric curve.

[00029] Disclosed is an example method for generating three-dimensional (3D) printing points for printing a 3D object. The method may include obtaining a Steiner patch that is part of a tessellation approximation of a three-dimensional object to be printed by a 3D printer, determining a slicing curve, the slicing curve being an intersection of a slicing plane and the Steiner patch in Euclidian space, determining a parametric curve of the slicing curve, the parametric curve having a non-degenerate conic classification, reclassifying the parametric curve as a degenerate conic classification,

sampling points along the parametric curve based upon the degenerate conic classification and generating 3D printing points in Euclidean space for the 3D object based upon the points sampled along the parametric curve.

[00030] Figure 1 is a block diagram schematically illustrating portions of an example 3D printing system 20. 3D printing system 20 carries out printing of a three-dimensional structure or object based upon data contained in a received object data file 22. The data contained in the object data file 22 defines a model of the three-dimensional object to be printed. System 20 comprises 3D printing points 3D printing point generator 26 and 3D printer 30.

[00031] 3D printing point generator 26 generates the individual 3D printing slice/layer points 31 and transmits points 31 to 3D printer 30. 3D printer utilizes points 31 to form the layers of the 3D object defined by the object data file 22. In one implementation, 3D printer 30 selectively solidifies portions of 3D build material, in the form of powder or particulates, at each of the 3D printing points. In another implementation, 3D printer 30 selectively deposits or jets build material at each of the 3D printing points to form the layer of the object being printed. In some implementations, the 3D printing points may further serve to define where release materials or sacrificial materials are to be selectively deposited or formed by 3D printer 30. In one implementation, 3D printing points 3D printing point generator 26 is provided in a separate computing unit or device as 3D printer 30. In another implementation, 3D printing points 3D printing point generator 26 is incorporated into a single unit along with 3D printer 30.

[00032] 3D printing point generator 26 comprises processor 32 and memory 34. Processor 32 comprises electronics or circuitry that carries out instructions provided by memory 34. Memory 34 comprises a non-transitory computer-readable medium that contains such instructions. As will be described hereafter, such instructions may direct processor 32 to generate 3D printing slice/layer points from Steiner patches or a NURQS model in a more



efficient manner. Various implementations of the instructions contained in memory 34 may direct the process 32 to reduce the computational burden while enhancing accuracy: (1) by limiting the sampling of points along a parametric curve to points between border points of the parametric curve; (2) by iteratively adding sampling points along the parametric curve in parametric space until the spacing between each pair of consecutive sample points in Euclidean space satisfies a predefined threshold; and/or (3) reclassifying a non-degenerate conic parametric curve as a degenerate conic parametric curve for sampling the points along the parametric curve.

[00033] Figure 2 is a flow diagram of an example method 100 that may be carried out by 3D printing point generator 26, wherein memory 34 contains instructions for directing processor 32 to carry out method 100. Method 100 may reduce the computational burden of generating 3D printing points using Steiner patches by limiting the sampling of points along a parametric curve to points between border points of the parametric curve.

[00034] As indicated by block 104, 3D printing point generator 26 obtains an individual Steiner patch of a mesh of Steiner patches, the Steiner patch being part of a tessellation approximation of a three-dimensional object to be printed by 3D printer 30. Figure 3A illustrates an example of a mesh of Steiner patches, or? a NURQS model, representing a three-dimensional object that may be printed. In the example illustrated in Figure 3A, the mesh of Steiner patches 128 approximates a sphere 130. As should be appreciated, any variety of three-dimensional objects may be represented by a greater or fewer number of such Steiner patches. Figure 3B illustrates an individual Steiner patch 128 of the sphere 130 of Figure 3A and control points 134 which are the base ingredients of a mathematical expression to produce, along with a parametric information, points of the Steiner patch 128. The NURQS model may be provided as part of the object data file 22 supplied to 3D printing point generator 26 or may itself be generated from a model in another format contained in object data file 22. For example, in one

implementation, the object data file 22 may contain a NURBS model from a computer aided design device, wherein 3D printing point generator 26 generates NURQS model of Steiner patches from the NURBS model in accordance with any of various existing and future developed techniques.

[00035] Upon obtaining the larger NURQS file, 3D printing point generator 26 proceeds by generating 3D printing points 31 for each individual Steiner patch of the NURQS model, the 3D printing points being sent to 3D printer 30 for printing the individual layer of the 3D object being printed. Figure 4A illustrates an example individual Steiner patch 210 in Euclidean space. As indicated by block 106, 3D printing point generator 26 determines a slicing curve, the slicing curve being an intersection of a slicing plane and the Steiner patch in Euclidean space. The slicing plane corresponds to at least a portion of the layer of the 3D object to be printed and for which the 3D printing points are being generated. During the course of generating 3D printing points for printing the entire 3D object, each individual Steiner patch may be intersected by multiple different slicing planes, each of the slicing planes corresponding to the different layers of the object being printed on a layer-by-layer basis by printer 30. Figure 4A illustrates an example slicing plane 212 and its associated slicing curve 214.

[00036] As indicated by block 108, 3D printing point generator 26 determines a parametric curve of the slicing plane. The parametric curve is a conic in parametric space and constitutes a pre-image of the slicing curve which is in Euclidean space. Figure 4B further illustrates an example parametric curve 216 corresponding to the slicing curve 214.

[00037] As indicated by block 110, 3D printing point generator 26 determines a classification of the parametric curve, such as parametric curve 216. Figure 5 illustrates various non-degenerate classifications of a parametric curve. As shown by Figure 5, the parametric curve comprises a cross-section of a cone, comprising a conic. Example non-degenerate

classifications for a parametric curve include a circle 150, an ellipse 152, a parabola 154 and a hyperbola 156. It should be noted that the illustrated example hyperbola appears on the upper half of a double cone and is not seen in Figure 5. In some circumstances, the parametric curve may have a degenerate classification such as parallel lines, concurrent lines, a single-line, coincident lines or a single point. As will be described hereafter, the classification of the parametric curve determines how points are sampled along the parametric curve.

[00038] As indicated by block 112, using the classification of the parametric curve, 3D printing point generator 26 determines the border points of the parametric curve. Figure 4B illustrates the example parametric curve 216 as having parametric border points 220-1 and 220-2 which correspond to border points 221-1 and 221-2 of slicing curve 214.

[00039] As indicated by block 114, based upon the classification of the parametric curve, 3D printing point generator 26 samples parametric points 228 along the parametric curve between the border points. By limiting the sampling of parametric points to those points on the parametric curve between the border points, computational resources are preserved.

[00040] As indicated by block 116, using points 228 sampled between the border points 220, 3D printing point generator 26 generates corresponding Euclidean points, the 3D printing points, in Euclidean space for the object to be printed. 3D printing point generator 26 converts the sample parametric points 228 in parametric space to corresponding points 231 along slice curve 214 in Euclidean space. Points 231 in Euclidean space serve as a 3D printing points 31 for the layer of build material that is to form portions of the Steiner patch approximating a portion of the object being printed. Upon being generated,, the 3D printing points 31 may be output to the 3D printer 30 for printing a slice of the object.

[00041] Figure 6 is a flow diagram of an example method 300 that may be carried out by 3D printing point generator 26, wherein memory 34 contains instructions for carrying out method 300. Method 300 may reduce the computational burden and enhance printing resolution of generating 3D printing points using Steiner patches by iteratively adding sampling points along the parametric curve in parametric space until the spacing between each pair of consecutive sample points in Euclidean space (the Euclidean points corresponding to the parametric points in parametric space) satisfies a predefined threshold. Blocks 304-310 of method 300 correspond to blocks 104-110 of method 100, respectively, and are described above.

[00042] As indicated by block 312 and illustrated by Figure 7A, upon determining the parametric curve and its classification, 3D printing point generator 26 begins sampling points along the parametric curve 216. In the example illustrated, 3D printing point generator 26 samples a first parametric point 228-1 and a second parametric point 228-2 along the parametric curve 216 between border points 220. Such sampling of points 228 is based upon the classification of the parametric curve 216. In one implementation, the sampling of parametric points 228-1 and 228-2 between border points 228 is random. In another implementation, the sampling points is done to equidistantly space points 228-1 and 228-2 from one another and from border points 220. In other implementations, the sampling of parametric points along parametric curve 216 in between border points 220 may be done in other fashions.

[00043] The sampled parametric points 228-1 and 228-2 have corresponding Euclidean points 231-1 and 231-2, respectively, on the slicing curve 214 in Euclidean space. However, the relative positions of the Euclidean points 231-1 and 231-2 are not necessarily the same as the relative positions of the parametric points 228-1 and 228-2 in parametric space. As indicated by block 314, 3D printing point generator 26 determines a Euclidean spacing 241 separating the Euclidean points 231-1 and 231-2. 3D printing point generator 26 may also determine a Euclidean spacing 243, along the

slicing curve 214, separating Euclidean point 231-1 from border point 221-1 and a Euclidean spacing 245, along slicing curve 214, separating Euclidean point 231-2 from border point 228-2.

[00044] As indicated by block 316, 3D printing point generator 26 compares each of the Euclidean spacings 241, 243 and 245 against a predefined spacing threshold. The spacing threshold may be based upon a target tolerance or degree of resolution that itself may be a function of the precision of the 3D printer 30.

[00045] As indicated by block 318, in response to the Euclidean spacing or distance separating Euclidean points 231-1 and 231-2 failing to satisfy the threshold, and as shown by Figure 8A, 3D printing point generator 26 samples a third parametric point 228-3 along the parametric curve 216 between the prior parametric points 228-1 and 228-2. In one implementation, location at which the third parametric point between parametric points 228-1 and 228-2 is randomly chosen. In another implementation, the location at which the third parametric point 228-3 between parametric points 228-1 and 228-2 is chosen so as to lie equidistantly between parametric points 228-1 and 228-2.

[00046] As indicated by block 320, 3D printing point generator 26 generates the 3D printing points in Euclidean space for the object to be printed based upon the first sample point 228-1, the second sample point 228-2 in the third sample point 228-3 along the parametric curve 216. In particular, 3D printing point generator 26 determines the Euclidean points 231-1, 231-2 and 231-3 that correspond to the parametric points 228-1, 228-2 and 228-3, respectively. Each of the Euclidean points 231-1, 231-2 and 231-3 serves as a 3D printing point for the particular slicing plane 212 and the particular Steiner patch 210.

[00047] The general process of sampling parametric points along the parametric curve 214, identifying the Euclidean points corresponding to the sampled parametric points, measuring or determining a Euclidean spacing between the Euclidean points correspond to the sample parametric points and sampling in additional parametric point between the two prior sample

parametric points may be repeated until the Euclidean spacing between each Euclidean point on the slicing curve 214 satisfies the predefined spacing threshold. For example, block 314-318 may be carried out with respect to parametric border point 220-1 and parametric point 228-1. In particular, 3D printing point generator 26 may compare the Euclidean spacing 243 (shown in Figure 7B) to the predefined threshold and insert an additional parametric point between the parametric border point 220-1 and the parametric point 228-1 should the Euclidean spacing fail to satisfy the predefined spacing threshold. 3D printing point generator 26 may likewise compare the Euclidean spacing 245 (shown in Figure 7B) against the predefined spacing threshold and insert an additional parametric point between the parametric border point 221-2 and the parametric point 228-2. For each of these inserted parametric points, a corresponding Euclidean point may be identified in Euclidean space and used to generate a corresponding 3D printing point.

[00048] The above described iterations may continue following each insertion or sampling of a new parametric point between two consecutive previously sampled parametric points until each Euclidean spacing (or a predetermined percentage of Euclidean spacings) between Euclidean points corresponding to the parametric points satisfy the predefined spacing threshold. For example, as shown by Figure 8B, following the insertion of the newly sampled parametric point 228-3 between parametric points 228-1 and 228-2, 3D printing point generator 26 determines the coordinates of the Euclidean point 231-3 on the slicing curve 214 that corresponds to the parametric point 228-3. 3D printing point generator 26 then determines the Euclidean spacings on either side of the newly inserted Euclidean point 231-3 separating the Euclidean point 231-3 from the next consecutive Euclidean point on slicing curve 214. In the example illustrated, 3D printing point generator 26 determines the Euclidean spacing 247 spacing Euclidean points 231-1 and 231-3 and Euclidean spacing 249 spacing Euclidean points 231-3 and 231-2. 3D printing point generator 26 then compares each of the

determine Euclidean spacings 247 and 249 against the predefined spacing threshold.

[00049] As shown by Figure 9A, in response to the Euclidean spacing 247 being greater than the predefined spacing threshold, 3D printing point generator 26 proceeds by inserting or sampling yet another parametric point 228-4 between the parametric points 228-1 and 228-3 which correspond to Euclidean points 231-1 and 231-3. In response to the Euclidean spacing 249 being greater than the predefined spacing threshold, 3D printing point generator 26 proceeds by inserting or sampling yet another parametric point 228-5 between the parametric points 228-3 and 228-2 which correspond to Euclidean points 231-3 and 231-2. As shown by Figure 9B, each of the parametric points 228-1, 228-2, 228-3, 228-4 and 228-5, as well as the border parametric points 220-1 and 220-2 has a corresponding Euclidean point along the slicing curve 214 which serves as a 3D printing point. As discussed above, this process iteratively continues until each determined Euclidean spacing between consecutive Euclidean points along the slicing curve 214 satisfy the predefined spacing threshold or until another Euclidean spacing criteria, such as a percentage of the locating spacings, is satisfied. Once all of the parametric points and their corresponding Euclidean points have been determined, and the 3D printing points corresponding to the Euclidean points have been generated, the 3D printing points may be used by 3D printer 30 to print a slice of the 3D object.

[00050] As shown by Figure 10A, in some implementations, 3D printing point generator 26 may sample a predefined number of initial parametric points between the border parametric points 220 before determining any Euclidean spacings for the parametric points. For example, 3D printing point generator 26 may initially sample parametric points 228-1 to 228-n between border parametric points 220, wherein n may be a number greater than two. For example, 3D printing point generator 26 may initially sample 100 parametric points 231 on the parametric curve 216.

[00051] As shown by Figure 10B, each of the parametric points 231 that are sampled have a corresponding Euclidean point on slicing curve 214. For each consecutive pair of Euclidean points on slicing curve 214, 3D printing point generator 26 may determine a Euclidean spacing between such Euclidean points. For every instance where the determined Euclidean spacing fails to satisfy the predetermined spacing threshold, 3D printing point generator 26 may proceed with sampling inserting a new parametric point between the parametric points that corresponded to the two Euclidean points separated by the Euclidean spacing that failed to satisfy the predetermined spacing threshold. As above, each insertion of a new parametric point results in the insertion of a new Euclidean point, wherein 3D printing point generator 26 determines new Euclidean spacings on each side of the new Euclidean point to determine whether the new Euclidean spacings satisfy the predetermined spacing threshold.

[00052] Figure 11 is a flow diagram of an example method 400 that may be carried out by 3D printing point generator 26, wherein memory 34 contains instructions for directing processor 32 to carry out method 100. Method 400 may reduce the computational burden of generating 3D printing points using Steiner patches by reclassifying a non-degenerate conic parametric curve as a degenerate conic parametric curve for sampling the points along the parametric curve. Blocks 404 and 406 correspond to blocks 104 and 106 of method 100 and are described above.

[00053] Block 408 is similar to block 108 described above except that the parametric curve determined by 3D printing point generator 26 in block 408 has a non-degenerate conic classification. In particular, the parametric curve has a conic classification selected from a group of classifications consisting of a circle 150, and ellipse 152, a parabola 154 and a hyperbola 156 as shown in Figure 12.

[00054] As indicated by block 409, 3D printing point generator 26 automatically reclassify the parametric curve as a degenerate conic classification. Although the actual conic shape of the parametric curve is that



of a non-degenerate conic, 3D printing point generator 26 treats the parametric curve as a degenerate conic fast occasion so as to reduce processing demands. As further shown by Figure 12, 3D printing point generator 26 analyzes the shape, characteristics and distances between various points of the parametric curve and based upon such analysis reclassifies the parametric curve as a degenerate, sampling the parametric points along the parametric curve using mathematical formula and compilations as if the parametric curve actually comprised parallel lines 162, concurrent lines 164, a single line 166, coincident lines 168 or a single point 170.

[00055] Figures 13A and 13B illustrate one example of a circumstance where 3D printing point generator 26 carries out method 402 reclassify an otherwise non-degenerate parametric curve with a degenerate conic classification. Figure 13A illustrates an example Steiner patch 610 intersected by an example slicing plane 612. As discussed above, Steiner patch 610 is part of a mesh or tessellation of Steiner patches (a NURQS model), wherein slicing plane 612 corresponds to portions of the individual layer of the 3D object being printed by 3D printer 30. Figure 13A further illustrates the example slicing curve 614 formed by the intersection of Steiner patch 610 and slicing plane 612.

[00056] Figure 13B illustrates the corresponding parametric curve 616 in parametric space. In the example illustrated, parametric curve 616 has a parabola conic classification. In response to determining that parametric curve 616 has a parabola conic classification, 3D printing point generator 26 evaluates selected points of permit curve 616 relative to portions of the parametric space to determine whether the particular parametric curve 616 should instead be reclassified or alternatively treated as a degenerate conic classification, such as a single line classification.

[00057] In one implementation, 3D printing point generator 26 determines a parametric distance of a point of the parametric curve 616 from an origin 619 of the parametric space 621. Generator 61 compares the

determine parametric distance to a predefined distance threshold. In response to the distance threshold being satisfied, being greater than a predetermined threshold, 3D printing point generator 26 classifies curve 616 as a single line classification. The single line classification is used to sample points along the parametric curve 616, wherein classifying points along a single line parametric curve may be less computationally burdensome as compared to sampling parametric points along a parametric curve having a parabola conic classification.

[00058] In one implementation, the point evaluated by 3D printing point generator 26 to determine the parametric distance comprises a center 623 of the parametric curve 616. 3D printing point generator 626 determines the parametric distance by comparing the coordinates of the center 623 and the origin 619. In response to the parametric distance satisfying the predefined distance threshold, 3D printing point generator 26 classifies curve 616 as a single line classification.

[00059] In another implementation, the point evaluated by 3D printing point generator 26 to determine the parametric distance comprises a vertex 625 of the parabola forming parametric curve 616. Generator 626 uses the coordinates of the vertex 625 relative to the coordinates of origin 619 to determine the parametric distance in response to the parametric distance satisfying the predefined distance threshold, 3D printing point generator 26 classifies curve 616 as a single line classification.

[00060] As shown by Figure 11, upon reclassifying the parametric curve, 3D printing point generator 26 proceeds with blocks 414 and 416. Blocks 414 in foreign 16 correspond to blocks 114 and one earned 16 of method 100 described above. As described above, 3D printing point generator 26 samples points along the parametric curve based upon the reclassification, the degenerate conic classification. In the example illustrated in Figures 13A and 13B, 3D printing point generator 26 samples parametric points along the parametric curve based upon the degenerate single line conic classification. As indicated by block 416, using the parametric points sampled in block 414,

3D printing point generator 26 generates Euclidean points, 3D printing points, in Euclidean space for the 3D object.

[00061] Each of the above described methods 100, 300 and 400 may be combined as part of a single overall 3D printing point generation method. For example, in one implementation, memory 34 of 3D printing point generator 26 may contain instructions for directing processor 32 to carry out a single overall method similar to method 300 described above except that the points are specifically sample between the parametric border points as described above with respect to method 100 and that block 310 comprises an evaluation of the parametric curve and possible reclassification of a parametric curve having a non-degenerate conic classification as a parametric curve having a degenerate conic classification, such as a single line classification as described above with respect to method 400.

[00062] Below is a description of the mathematical foundations and computations that may be carried out by 3D printing point generator 26 in accordance with instructions contained in memory 34 when generating 3D printing points in accordance with the above described methods. It should be appreciated that the above described methods may be carried out using other mathematical computations or other transformations of Steiner patches provided in or derived from the data contained in the object data file 22. As noted above, A Steiner Patch in rational form as a function of parameters (s, t), given the control points  $P_{20}, P_{02}, P_{10}, P_{01}, P_{11}$  and  $P_{00}$ , along with their intermediary weights  $w_{10}, w_{01}$  and  $w_{11}$ , described as follows:

$$S(s, t) = \frac{A_{20}s^2 + A_{02}t^2 + A_{11}st + A_{10}s + A_{01}t + P_{00}}{D_{20}s^2 + D_{02}t^2 + D_{11}st + D_{10}s + D_{01}t + 1} \quad (Eq. 1)$$

where:

- $A_{20} = P_{20} + P_{00} - 2w_{10}P_{10}$  and  $D_{20} = 2 - 2w_{10}$
- $A_{02} = P_{02} + P_{00} - 2w_{01}P_{01}$  and  $D_{02} = 2 - 2w_{01}$
- $A_{10} = 2w_{10}P_{10} - 2P_{00}$  and  $D_{10} = 2w_{10} - 2$
- $A_{01} = 2w_{01}P_{01} - 2P_{00}$  and  $D_{01} = 2w_{01} - 2$
- $A_{11} = 2P_{00} + 2w_{11}P_{11} - 2w_{01}P_{01} - 2w_{10}P_{10}$  and  $D_{11} = 2 + 2w_{11} - 2w_{10} - 2w_{01}$ .

[00063] In order for the slicing process to occur, an intersection between such a patch and a plane parallel to the printing bay of the 3D printer is utilized. This process is described by the implicit equation:  $z = k_p$ , where  $k_p$  designates a height of a certain layer, measured from the printing bay up. The intersection can be calculated by just taking the z-component of Eq. 1 and equating to  $k_p$ . If we let the z-component of  $A_{ij}$  to be  $z_{ij}$ , then the intersection yields the following equation on s and t parameters:

$$(z_{20} - D_{20}k_p)s^2 + (z_{11} - D_{11}k_p)st + (z_{02} - D_{02}k_p)t^2 + (z_{10} - D_{10}k_p)s + (z_{01} - D_{01}k_p)t + (z_{00} - k_p) = 0$$

[00064] This equation can be further simplified by renaming the s and t terms' multipliers with  $C_1, C_2, \dots, C_5$  and naming the independent term

$(z_{00} - k_p) = C_6$ , thus making

$$C_1 s^2 + C_2 st + C_3 t^2 + C_4 s + C_5 t + C_6 = 0 \quad (Eq. 2)$$

[00065] Equation 2 describes a conic curve in parametric space, which is the pre-image of the intersection between the Steiner Patch and the plane (in the Euclidean space). To properly sample points in this curve, a determination is made as to which type of conic this equation refers to, and that means that a new system of coordinates (with new variables  $s'$  and  $t'$ ) in the parametric space should be established in a such a way that the mixed term  $s't'$  does not show up in the new equation.

[00066] In Equation 2 the  $C_2st$  term being non zero is an indication that the new axes of the variables  $s'$  and  $t'$  are to be rotated of a certain angle  $\alpha_s$  in relation to their counterparts s and t in order for the mixed term  $s't'$  to become nullified in the yielded equation. The angle  $\alpha_s$  is already known in Analytical Geometry as being such that  $\tan(2\alpha_s) = \frac{C_2}{C_1 - C_3}$ , in case  $C_1 - C_3 \neq 0$ , otherwise  $\alpha_s = 45^\circ$ . This is analogous to rotating the parametric space of the Steiner patch such that it aligns with the canonical axis as shown in Figure 3. The rotation of  $\alpha_s$  produces the relations:  $s = s' \cos(\alpha_s) - t' \sin(\alpha_s)$  and  $t = s' \sin(\alpha_s) + t' \cos(\alpha_s)$  that can be plugged into Equation 2 to generate the following reduced equation of a conic in the variables  $s'$  and  $t'$ :

$$K_1 s'^2 + K_2 t'^2 + K_3 s' + K_4 t' + C_6 = 0 \quad (Eq. 3)$$

where

$$K_1 = C_1 \cos^2 \alpha_s + C_2 \cos \alpha_s \sin \alpha_s + C_3 \sin^2 \alpha_s$$

$$K_2 = C_1 \cos^2 \alpha_s - C_2 \cos \alpha_s \sin \alpha_s + C_3 \sin^2 \alpha_s$$

$$K_3 = C_4 \cos \alpha_s + C_5 \sin \alpha_s$$

$$K_4 = -C_4 \sin \alpha_s + C_5 \cos \alpha_s$$

[00067] With Equation 3 it is possible to classify the type of conic which is in the parametric space the pre-image of the slice curve that comes from the intersection between a Steiner patch and a plane. First, a check is made as to whether this is a degenerate conic (single point, intersecting lines, parallel lines and coincident lines). Consider the following symmetric matrix:

$$M = \frac{1}{2} \begin{pmatrix} 2C_1 & C_2 & C_4 \\ C_2 & 2C_3 & C_5 \\ C_4 & C_5 & 2C_6 \end{pmatrix}$$

[00068] If the determinant of M is zero, then the conic is degenerate. Otherwise, in the case of non-degeneracy, if on top of this  $K_1$  and  $K_2$  are both nonzero and present the same sign, then Equation 2 represents an Ellipse and If they present opposite signs, then Equation 2 represents a hyperbola. If either  $K_1$  or  $K_2$  are zero, then Equation 2 represents a parabola. Each type of conic uses its own form of parametrization, which has also to take into consideration the parametrization speed, in order to produce an equally spaced sampling, or at least as close as possible to this.

[00069] Unless explicitly defined otherwise, the calculation of the patch-plane intersection curve is a conic in parametric space which is further parameterized in terms of an angle  $\theta_s$  covering the full length of the conic. Since, generally, a portion of the conic is contained within the parametric space of the patch, this angle  $\theta_s$  will be bounded by the inferior and superior limits of the border points. The general algorithm for slice computation comprises finding the appropriate parameterization in terms of  $\theta_s$ , finding the points where the slice intersects with the patch's border, finding the

correspondent parameters of  $\theta_s$  for the border points and sampling the slice for the valid interval of  $\theta_s$ . The computed points are yielded in terms of the transformed parametric space  $(s', t')$ . A final transformation to  $(s, t)$  is used and can be computed through

$$s = s' \cos(\alpha_s) - t' \sin(\alpha_s), t = s' \sin(\alpha_s) + t' \cos(\alpha_s).$$

**[00070] General Notation**

**[00071]** The detailed process of computing the slices is broken down for each classification of the conic that is the pre-image of the intersection between a Steiner patch and a plane. These are the non-degenerate cases of the ellipse, parabola, hyperbola and the degenerate cases of intersecting lines, parallel lines, single line and point. There is also a special case – treated separately - where the conic arc segment is so far from the center that it is approximately straight for the portion that intersects with the patch.

**[00072]** To simplify the calculations for obtaining the slices, it is useful to define some previous notations.

$$\text{Let } G_1 = \frac{1}{|K_1|} \left( \frac{1}{4} \frac{K_3^2}{K_1} + \frac{1}{4} \frac{K_4^2}{K_2} - K_5 \right), G_2 = \frac{1}{|K_2|} \left( \frac{1}{4} \frac{K_3^2}{K_1} + \frac{1}{4} \frac{K_4^2}{K_2} - K_5 \right), A_s = \sqrt{|G_1|}, A_t = \sqrt{|G_2|}, s'_0 = \frac{-K_3}{2K_1} \text{ and } t'_0 = \frac{-K_4}{2K_2}.$$

**[00073] Ellipse Slicing**

**[00074]** As will be recurrent, the slicing method will first determine the parameter value  $\theta_s^b$  for the border point and then iterate  $\theta_s$  over the valid range. The parameter value  $\theta_s^b$  can be found by inputting the values of  $(s_b', t_b')$  (transformed from the point's corresponding  $(s, t)$  parameters) for a border point  $b$  in the following equations:

$$\theta_s^b = \begin{cases} \arccos\left(\frac{s_b' - s'_0}{A_s}\right), & \text{if } \arcsin\left(\frac{t_b' - t'_0}{A_t}\right) < 0 \\ 2\pi - \arccos\left(\frac{s_b' - s'_0}{A_s}\right), & \text{otherwise} \end{cases}.$$

**[00075]** Once the bounds for  $\theta_s$  are defined, the actual slice can be computed by evaluating the parameter within the specified range such that

$$s' = s'_0 + A_s \cos(\theta_s), t' = t'_0 + A_t \sin(\theta_s).$$

**[00076] Hyperbola Slicing**

[00077] Hyperbola slicing differs from the ellipse slicing due to the caveat that there are potentially two segments which are accounted for. For this type of conic, it is useful to separate the calculations of the border point parameter  $\theta_s^b$  by the quadrant in which the arc segment is located. The conditions for the value of  $\theta_s^b$  are given in the table below and will depend on the values of  $K_{cos}$  and  $K_{tan}$  computed as:

$$K_{cos} = \begin{cases} \frac{A_s}{s_b' - s_0'}, & \text{if } G_2 < 0 \\ \frac{A_t}{t_b' - t_0'}, & \text{if } G_1 < 0 \end{cases}, K_{tan} = \begin{cases} \frac{t_b' - t_0'}{A_t}, & \text{if } G_2 < 0 \\ \frac{s_b' - s_0'}{A_s}, & \text{if } G_1 < 0 \end{cases}$$

Quadrant	$\arccos(K_{cos})$	$\arctan(K_{tan})$	$\theta_s^b$
First	$< \frac{\pi}{2}$	$> 0$	$\arccos(K_{cos})$
Second	$\geq \frac{\pi}{2}$	$> 0$	$\arctan(K_{tan}) + \pi$
Third	$\geq \frac{\pi}{2}$	$\leq 0$	$\arccos(K_{cos})$
Fourth	$< \frac{\pi}{2}$	$\leq 0$	$\arctan(K_{tan}) + \pi$

[00078] Once the limits for  $\theta_s$  have been established, the slicing can be evaluated from the  $(s', t')$  parameters computed as:

$$s' = \begin{cases} s_0' + A_s \cos^{-1}(\theta_s), & \text{if } G_2 < 0 \\ s_0' + A_s \tan(\theta_s), & \text{otherwise} \end{cases}, t' = \begin{cases} t_0' + A_t \tan(\theta_s), & \text{if } G_2 < 0 \\ t_0' + A_t \cos^{-1}(\theta_s), & \text{otherwise} \end{cases}$$

**[00079] Parabola Slicing**

[00080] For the parabola-type slicing, a new definition for  $s_0'$  and  $t_0'$  is given. Besides, a new coefficient of  $dv$  is also defined. Have  $s_0' =$

$$\frac{1}{K_3} \left( K_5 - \frac{K_4^2}{4K_2} \right), t_0' = \frac{-K_4}{2K_2} \text{ and } dv = \frac{-K_3}{4K_2} \text{ if } K_1 = 0 \text{ or } s_0' = \frac{-K_3}{4K_2}, t_0' =$$

$$\frac{1}{K_4} \left( K_5 - \frac{K_3^2}{4K_1} \right) \text{ and } dv = \frac{-K_4}{2K_1} \text{ otherwise. Once again, the values for the}$$

parameters  $\theta_s^b$  can be computed from the transformed  $(s'_b, t'_b)$  parameters of the border points. Make

$$\theta_s^b = \begin{cases} \frac{t'_b - t'_0}{2dv}, & \text{if } K_1 = 0 \\ \frac{s'_b - s'_0}{2dv}, & \text{otherwise} \end{cases}.$$

[00081] Once  $\theta_s^b$  has been defined, the values for  $(s', t')$  of the slicing, points can be obtained by varying  $\theta_s$  within the valid interval such that:

$$s' = \begin{cases} \theta_s^2 dv + s'_0, & \text{if } K_1 = 0 \\ 2\theta_s dv + s'_0, & \text{otherwise} \end{cases}, t' = \begin{cases} 2\theta_s dv + t'_0, & \text{if } K_1 = 0 \\ \theta_s^2 dv + t'_0, & \text{otherwise} \end{cases}.$$

[00082] **Degenerate Conic Curves**

[00083] Let

$$\zeta = \det \begin{pmatrix} C_1 & \frac{1}{2}C_3 & \frac{1}{2}C_4 \\ \frac{1}{2}C_3 & C_2 & \frac{1}{2}C_5 \\ \frac{1}{2}C_4 & \frac{1}{2}C_5 & C_6 \end{pmatrix}, \quad \lambda = \det \begin{pmatrix} C_1 & \frac{1}{2}C_3 \\ \frac{1}{2}C_3 & C_2 \end{pmatrix},$$

$$\Omega = \det \begin{pmatrix} C_1 & \frac{1}{2}C_4 \\ \frac{1}{2}C_4 & C_6 \end{pmatrix}.$$

[00084] A conic that is the pre-image of the intersection between a Steiner Patch and a plane is considered degenerate if  $\zeta = 0$ . A further classification is given by the values of  $\lambda$  and  $\Omega$ . If  $\lambda > 0$  the conic is reduced to a point, and if  $\lambda < 0$  it is reduced to a pair of intersecting lines. In the case of  $\lambda = 0$ , the conic will be classified as coincident lines if  $\Omega = 0$ , or parallel lines if  $\Omega < 0$ .

[00085] **Intersecting Lines Slicing**

[00086] The conic image will be classified as the degenerate type of intersecting lines if either  $C_1$  or  $C_2$  are zero. As this conic type is composed of two independent line segments which intersect at a point, it is natural to first identify the parameters for the limits of the valid region of the segments within



the patch and then iterate over the segment. The value of the valid  $(s, t)$  parameters for the points will depend on whether  $C_1$  or  $C_2$  is zero. Since this is not parameterized as a conic but rather as a line segment, the parameters for the limits of the conic are defined simply as the  $(s, t)$  values of the border point. From the coefficients of the curve, it is possible to obtain  $x_0$  and  $y_0$ .

$$y_0 = \frac{2C_1C_5 - C_3C_4}{4C_1C_2 - C_3^2}, \quad x_0 = \begin{cases} \frac{C_4 - C_3y_0}{2C_1}, & \text{if } C_1 \neq 0 \\ \frac{C_5 - 2C_2y_0}{C_3}, & \text{otherwise} \end{cases}$$

[00087] The calculation for the  $(s, t)$  parameter pair for the line segments will be achieved by expressing  $s$  in terms of  $t$  or vice-versa. There will be two equations, one for each segment. A straightforward approach is to parameterize  $s$  in terms of  $t$  by the pairs of equations  $s = \frac{(-C_3 + \alpha_s)(t + y_0)}{2C_1} - x_0$  and  $s = \frac{(-C_3 - \alpha_s)(t + y_0)}{2C_1} - x_0$ .

[00088] However, there is a possibility of  $C_1 = 0$ , breaking the equation. In that case,  $y_0 = \frac{2C_1C_5 - C_3C_4}{4C_1C_2 - C_3^2}$  simplifies to  $y_0 = \frac{C_4}{C_3}$  and  $t$  is placed in terms of  $s$  instead resulting in the following pair of equations:  $t = \frac{-C_3(s + x_0)}{C_2 - y_0}$  and  $t = -y_0$  for all  $s$  within the interval.

[00089] One should consider the possibility of  $C_2 = 0$  along with  $C_1$ . In this case, one segment will be given in terms of  $s = \frac{-C_5}{C_3}$  for all valid  $t$  and the other given in terms of  $t = \frac{-C_4}{C_3}$  for all valid  $s$ .

#### [00090] Parallel Lines Slicing

[00091] If the conic curve in the parametric space degenerates as a pair of parallel lines, then each segment is computed independently. Let  $\delta = C_4^2 - 4C_1C_6$ . If  $C_1 \neq 0$  and  $\delta \geq 0$ , then the values of  $s$  can be parameterized in terms of  $t$  for both segments:

[00092] Let  $s_1 = \frac{-C_3t_1 - C_4 + \sqrt{\delta}}{2C_1}$  for all valid  $t_1$  on the first segment and let  $s_2 = \frac{-C_3t_2 - C_4 - \sqrt{\delta}}{2C_1}$  for all valid  $t_2$  on the second segment. If, however,  $C_1 = 0$  but

$C_2 \neq 0$ , then instead make  $\delta = C_5^2 - 4C_2C_6$  and compute  $t_1 = \frac{-C_5 + \delta}{2C_2}$  constant for all valid  $s_1$  in the first segment and  $t_2 = \frac{-C_5 - \delta}{2C_2}$  also constant for every valid  $s_2$  on the second segment. If none of the conditions are met the parameter pair will either be  $\left(\frac{-C_6}{C_4}, t\right)$  for all valid  $t$  if  $C_4 > C_5$  or  $\left(s, \frac{-C_6}{C_5}\right)$  otherwise.

**[00093] Single Line Slicing**

**[00094]** For the case where the conic curve has degenerated into a single line, the parameters for the slice will be given by  $s = \frac{-C_5t - C_6}{C_4}$  for all valid  $t$  if  $C_4 \neq 0$ . If  $C_4 = 0$  however, then the slicing will be put in terms of  $t = \frac{-C_6}{C_5}$ , with  $s$  arbitrary within the interval.

**[00095] Coincident Lines Slicing**

**[00096]** In some instances, the conic curve in the parametric space will degenerate into a pair of coincident lines. Normally this case would be like the single line, however since these are limited line segments are treated specifically. Since there are several possible parameterizations for the values of  $(s, t)$  depending on the values of  $C_1, C_2, \dots, C_6$ , it is useful to display them as a table:

$C_3 \neq 0$	$C_6 = \frac{C_4C_5}{2C_3}$	$C_3 > 0$	$\sqrt{C_1}s + \sqrt{C_2}t = -\sqrt{C_6}$ or $\sqrt{C_1}s + \sqrt{C_2}t = \sqrt{C_6}$
		$C_3 < 0$	$\sqrt{C_1}s - \sqrt{C_2}t = \sqrt{C_6}$ or $\sqrt{C_1}s - \sqrt{C_2}t = -\sqrt{C_6}$
$C_3 = 0$	$C_1 = 0, \quad C_2 \neq 0$	$C_6 = \frac{C_5^2}{4C_2}$	$t = \frac{-C_5}{2C_2}$ for all $s$
	$C_1 \neq 0, \quad C_2 = 0$	$C_6 = \frac{C_4^2}{2C_1}$	$s = \frac{-C_4}{2C_1}$ for all $t$

**[00097] Point Slicing**

[00098] Computing the slice for the conic which degenerated into a point

is trivially accomplished by letting  $s = \frac{c_3 c_4 - 2c_1 c_5}{4c_1 c_2 - c_3^2}$ ,  $t = \begin{cases} \frac{c_3 y_0 - c_4}{2c_1}, & \text{if } c_1 \neq 0 \\ \frac{2c_2 y_0 - c_5}{c_3}, & \text{otherwise} \end{cases}$ .

[00099] Although the present disclosure has been described with reference to example implementations, workers skilled in the art will recognize that changes may be made in form and detail without departing from disclosure. For example, although different example implementations may have been described as including features providing various benefits, it is contemplated that the described features may be interchanged with one another or alternatively be combined with one another in the described example implementations or in other alternative implementations. Because the technology of the present disclosure is relatively complex, not all changes in the technology are foreseeable. The present disclosure described with reference to the example implementations and set forth in the following claims is manifestly intended to be as broad as possible. For example, unless specifically otherwise noted, the claims reciting a single particular element also encompass a plurality of such particular elements. The terms “first”, “second”, “third” and so on in the claims merely distinguish different elements and, unless otherwise stated, are not to be specifically associated with a particular order or particular numbering of elements in the disclosure.

WHAT IS CLAIMED IS:

1. A non-transitory computer-readable medium containing instructions to direct a processor to:

- obtain a Steiner patch that is part of a tessellation approximation of a three-dimensional (3D) object to be printed by a 3D printer;
- determine a slicing curve, the slicing curve being an intersection of a slicing plane and the Steiner patch in Euclidian space;
- determine a parametric curve of the slicing curve, the parametric curve existing in a parametric space;
- determine a classification of the parametric curve;
- determine border points of the parametric curve;
- sample points along the parametric curve between the border points of the parametric curve based on the classification; and
- generate 3D printing points in Euclidean space for the object based upon the sampled points, the 3D printing points for use in 3D printing of the 3D object.

2. The medium of claim 1, wherein the sampling of points along the parametric curve comprises sampling a first point and sampling a second point spaced from the first point by a parametric spacing along the parametric curve and between the border points of the parametric curve and wherein the instructions are to further direct the processor to:

6                    determine a Euclidean spacing of the first point and the  
7                    second point in Euclidean space;

8                               compare the Euclidean spacing to a predefined spacing  
9                               threshold; and

10                              in response to the Euclidean spacing failing to satisfy the  
11                              predefined threshold, sample a third point along the parametric  
12                              curve between the first point and the second point in parametric  
13                              space, wherein the 3D printing points are generated based upon  
14                              the third point.

15               3.       The medium of claim 1, wherein the sampling of points along the  
16                      parametric curve comprises sampling a first point and sampling a second point  
17                      spaced from the first point by a parametric spacing along the parametric curve  
18                      and between the border points in parametric space, the instructions are to further  
19                      direct the processor to:

20                              determine a Euclidean spacing of the first point and the  
21                              second point in Euclidean space;

22                              compare the Euclidean spacing to a predefined spacing  
23                              threshold; and

24                              in response to the Euclidean spacing failing to satisfy the  
25                              predefined threshold, sample a third point along the parametric  
26                              curve between the first point and the second point;

27                              determine a Euclidean spacing of the first point and the third  
28                              point in Euclidean space;

29                              compare the Euclidean spacing of the first point and the third  
30                              point to the predefined spacing threshold; and

31 in response to the Euclidean spacing of the third point and  
32 the first point not satisfying the predefined threshold,

33 sample a fourth point along the parametric curve between  
34 the first point and the third point;

35 determine a Euclidean spacing of the first point and the  
36 fourth point in Euclidean space;

37 compare the Euclidean spacing of the first point and the  
38 fourth point to the predefined spacing threshold; and

39 in response to the Euclidean spacing of the first point and  
40 the fourth point satisfying the predefined threshold, generate the 3D  
41 printing points based on the fourth point.

1 4. The medium of claim 1, wherein the instructions are to further direct  
2 the processor to:

3 in response to the parametric curve being classified  
4 as parabola,

5 compare vertex coordinates of the parametric  
6 curve to a distance threshold; and

7 in response to any of the vertex coordinates  
8 failing to satisfy the distance threshold, assign the  
9 parametric curve a single line classification, wherein  
10 the sampling of points along the parametric curve  
11 between the border points of the parametric curve is  
12 based upon the single line classification.

13           5.       The medium of claim 1, wherein the instructions are to further direct  
14 the processor to replace a non-degenerate classification of the parametric curve  
15 with a degenerate classification of the parametric curve, wherein sampling of  
16 points along the parametric curve between the border points of the parametric  
17 curve is based upon the degenerate classification.

18           6.       The medium of claim 1, wherein the instructions are to further direct  
19 the processor to:

20                       determine a parametric distance of a point of the parametric  
21 curve from an origin of the parametric space;

22                       assign the parametric curve a single line classification in  
23 response to the parametric distance of the point of the parametric  
24 curve satisfying a predefined distance threshold, wherein the  
25 sampling of points along the parametric curve between the border  
26 points of the parametric curve is based upon the single line  
27 classification.

28           7.       The medium of claim 6, wherein the point of the parametric curve  
29 used for determining the parametric distance is a center of the parametric curve.

30           8.       The medium of claim 6, wherein the parametric curve has a  
31 parabola classification and wherein the point of the parametric curve used for  
32 determining the parametric distance is a vertex of the parametric curve.

33           9.       A computer implemented method for generating three-dimensional  
34 (3D) printing points for printing a 3D object, the method comprising:

35                       obtaining a Steiner patch that is part of a tessellation  
36 approximation of the 3D object;

37                   determine a slicing curve, the slicing curve being an  
38                   intersection of a slicing plane and the Steiner patch in Euclidean  
39                   space;

40                   determine a parametric curve of the slicing curve, the  
41                   parametric curve existing in a parametric space;

42                   determining a classification of the parametric curve;

43                   sampling a first point and sampling a second point spaced  
44                   from the first point by a parametric spacing along the parametric  
45                   curve based upon the classification;

46                   determining a Euclidean spacing of the first point and the  
47                   second point;

48                   comparing the Euclidean spacing to a predefined spacing  
49                   threshold; and

50                   in response to the Euclidean spacing failing to satisfy the  
51                   predefined threshold, sampling a third point along the parametric  
52                   curve between the first point and the second point in parametric  
53                   space;

54                   generate 3D printing points in Euclidean space for the object  
55                   based upon the first point, second point and third point sampled  
56                   along the parametric curve, the 3D printing points for use in 3D  
57                   printing of the 3D object.

58

1           10.   The method of claim 9 further comprising:



2                   determine a Euclidean spacing of the first point and the third  
3                   point;  
4                   compare the Euclidean spacing of the first point and the third  
5                   point to the predefined spacing threshold; and  
6                   in response to the Euclidean spacing of the third point and  
7                   the first point not satisfying the predefined threshold,  
8                   sample a fourth point along the parametric curve between  
9                   the first point and the third point;  
10                  determine a Euclidean spacing of the first point and the  
11                  fourth point;  
12                  compare the Euclidean spacing of the first point and the  
13                  fourth point to the predefined spacing threshold; and  
14                  in response to the Euclidean spacing of the first point and  
15                  the fourth point not satisfying the predefined threshold, sample a  
16                  fifth point along the parametric curve between the first point and the  
17                  fourth point, wherein the generation of the 3D printing points in  
18                  Euclidean space for the object is based upon the first point, second  
19                  point, the third point, the fourth point and the fifth point sampled  
20                  along the parametric curve.

1           11.    The method of claim 11, wherein the parametric curve is assigned  
2           a single line classification, wherein the sampling of the first point and the second  
3           point along the parametric curve is based upon the single line classification.

1           12.    The method of claim 11, wherein the parametric curve has a non-  
2 degenerate classification, the method comprising sampling the first point and the  
3 second point along the parametric curve based upon a degenerate classification.

1           13.    A method for generating three-dimensional (3D) printing points for  
2 printing a 3D object, the method comprising:

3                   obtaining a Steiner patch that is part of a tessellation  
4 approximation of a three-dimensional object to be printed by a 3D  
5 printer;

6                   determine a slicing curve, the slicing curve being an  
7 intersection of a slicing plane and the Steiner patch in Euclidian  
8 space;

9                   determine a parametric curve of the slicing curve, the  
10 parametric curve having a non-degenerate conic classification;

11                  reclassifying the parametric curve as a degenerate conic  
12 classification; and

13                  sampling points along the parametric curve based upon the  
14 degenerate conic classification; and

15                  generate 3D printing points in Euclidean space for the 3D  
16 object based upon the points sampled along the parametric curve.

17           14.    The method of claim 13 wherein the reclassifying of the parametric  
18 curve as a degenerate conic classification comprises reclassifying the parametric  
19 curve as a single line degenerate conic classification in response to the  
20 parametric distance of a point of the parametric curve satisfying a predefined  
21 distance threshold.

- 22           15.    The method of claim 14, wherein the sampling of points along  
23   parametric curve comprises iteratively adding sample points along the parametric  
24   curve until a spacing between each consecutive pair of the sample points in  
25   Euclidean space satisfies a predefined threshold.

1/6

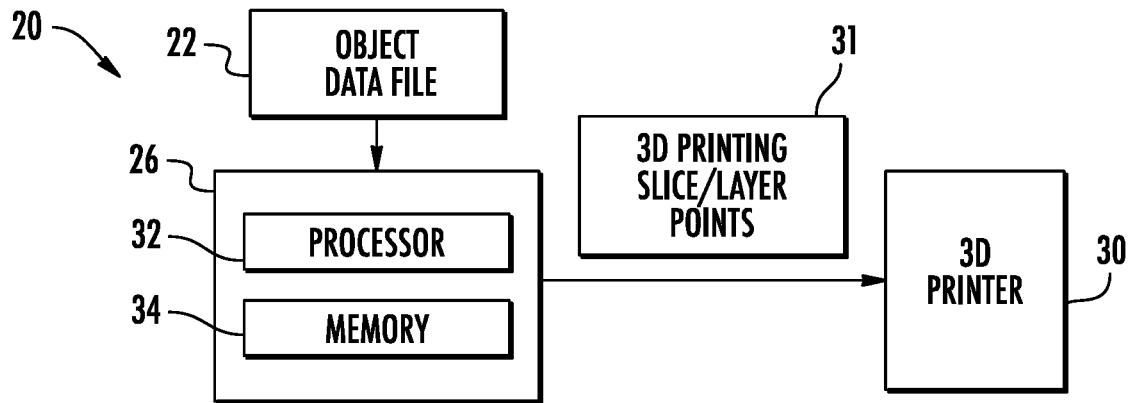


FIG. 1

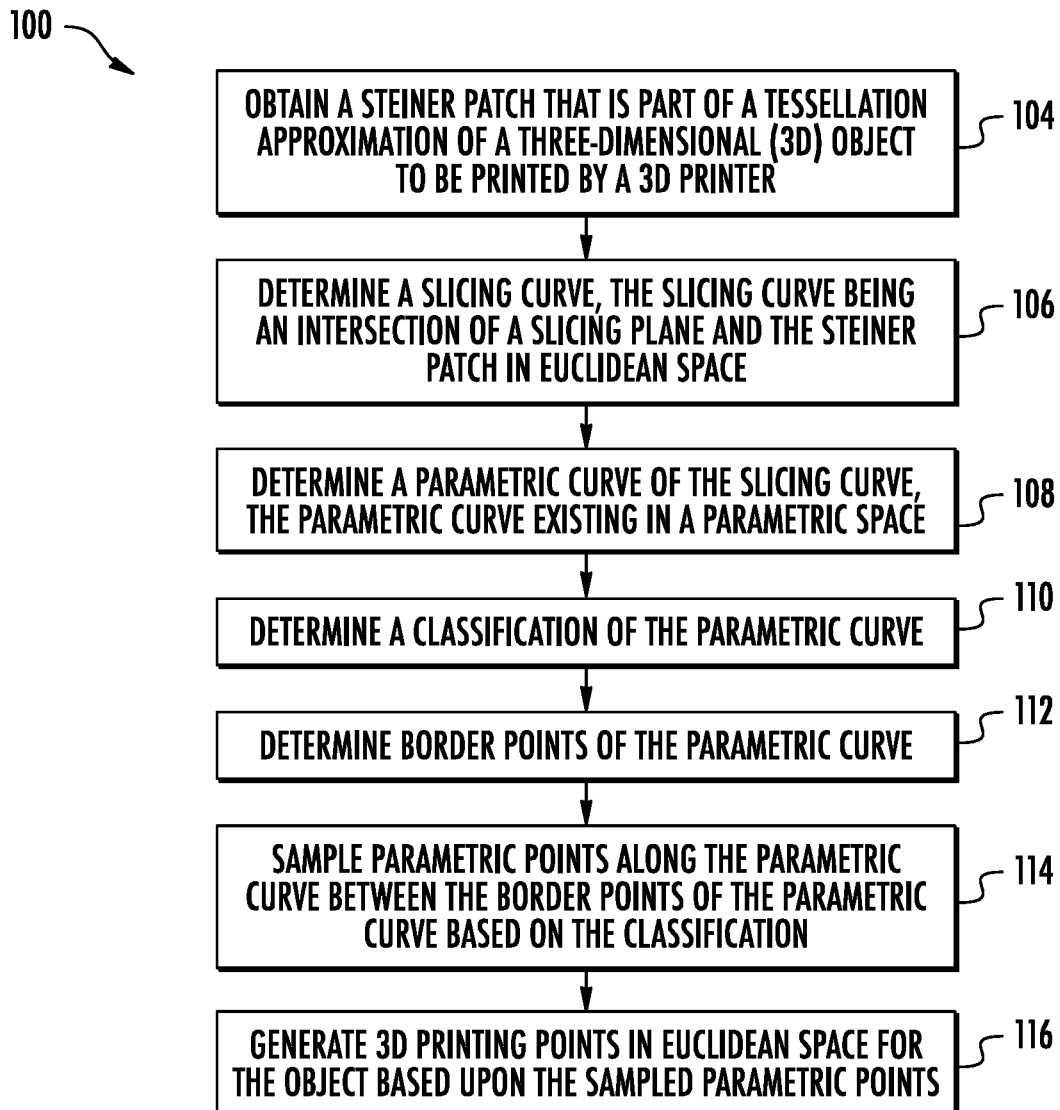
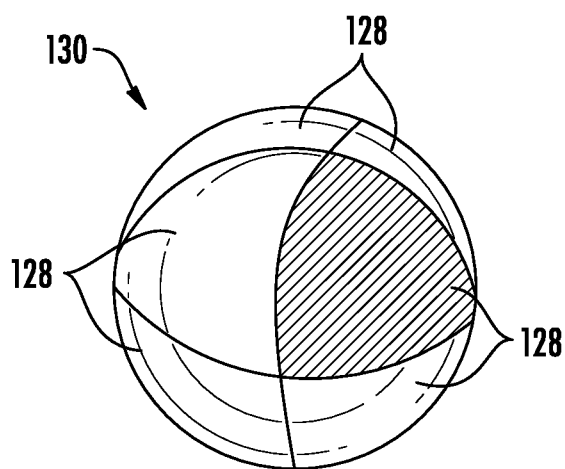
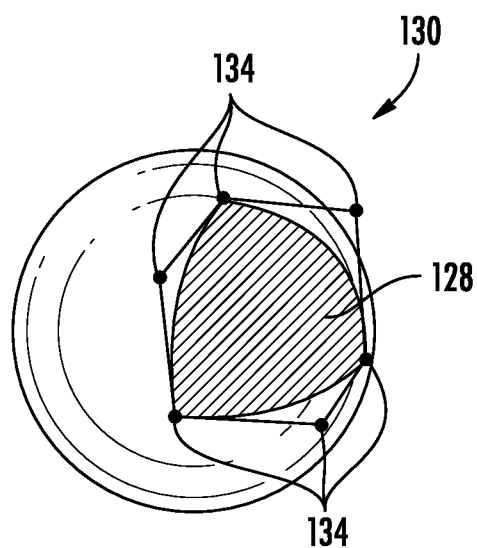


FIG. 2

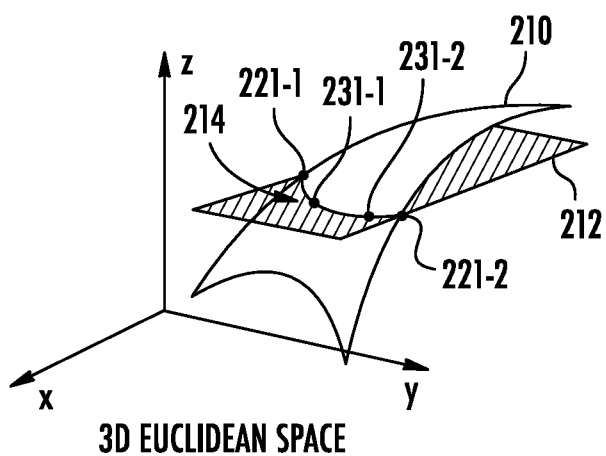
2/6



**FIG. 3A**

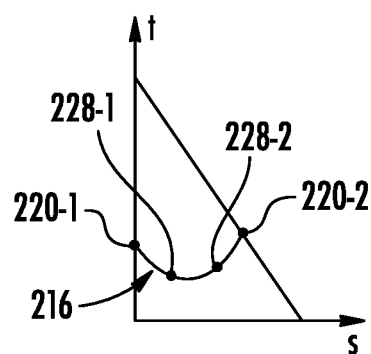


**FIG. 3B**



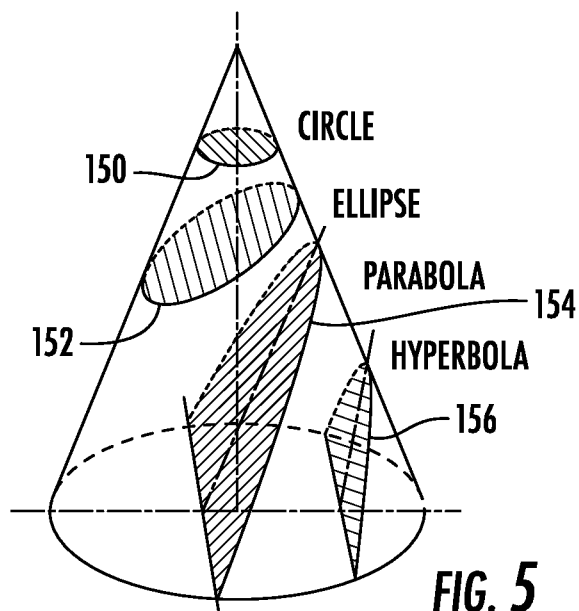
3D EUCLIDEAN SPACE

**FIG. 4A**



2D PARAMETRIC SPACE

**FIG. 4B**



**FIG. 5**

3/6

300

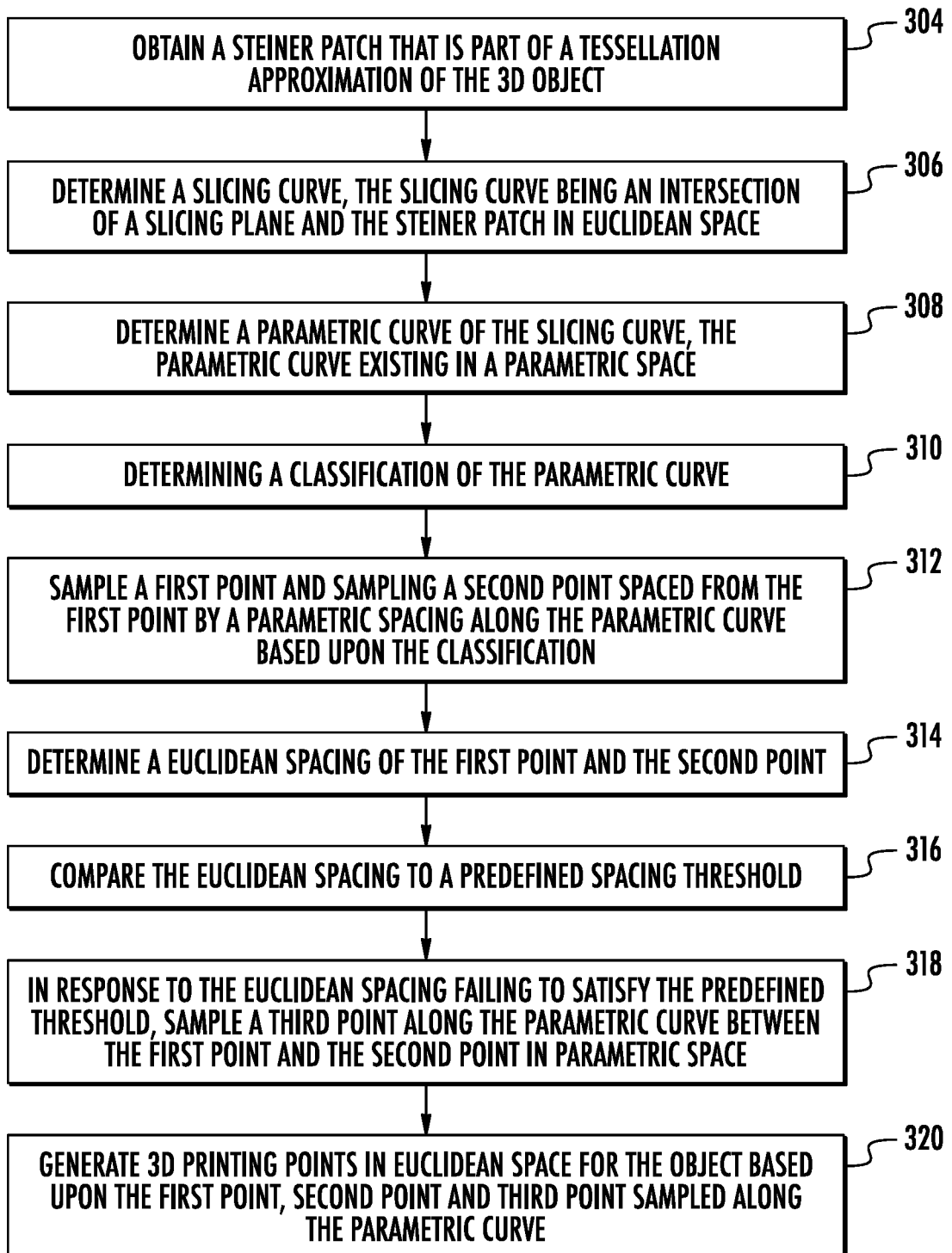
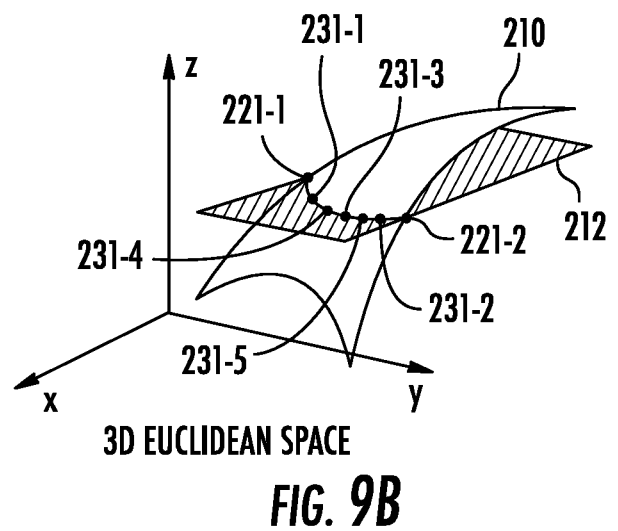
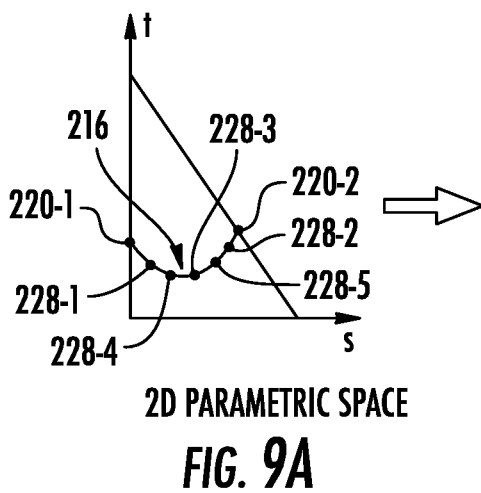
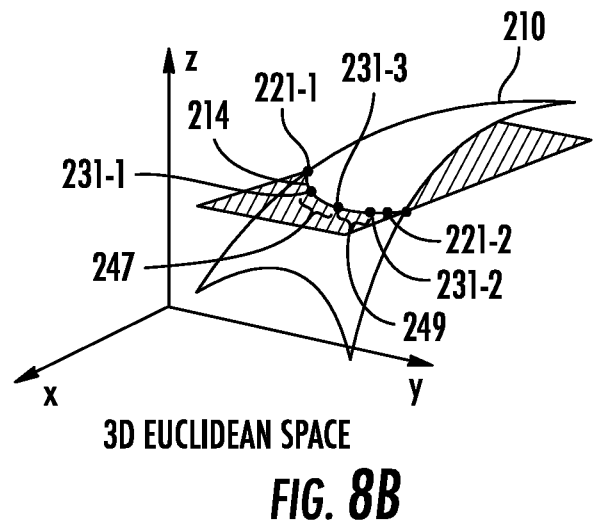
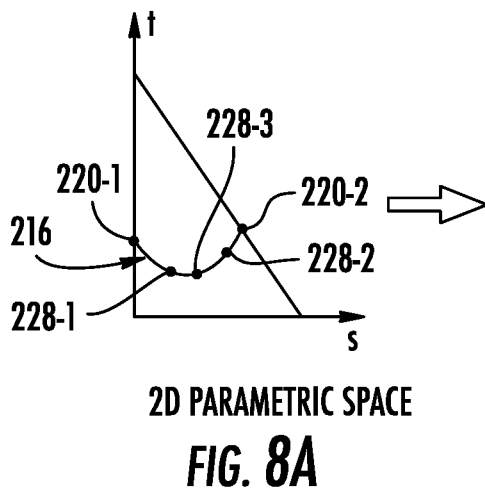
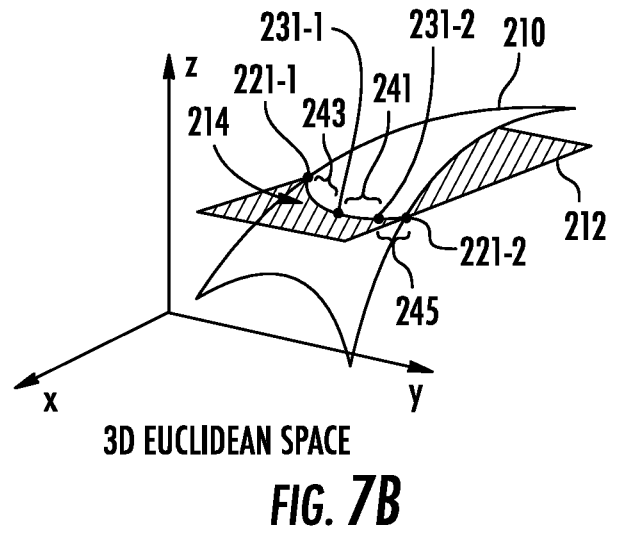
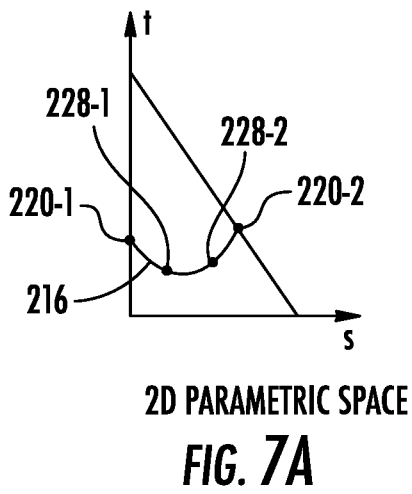
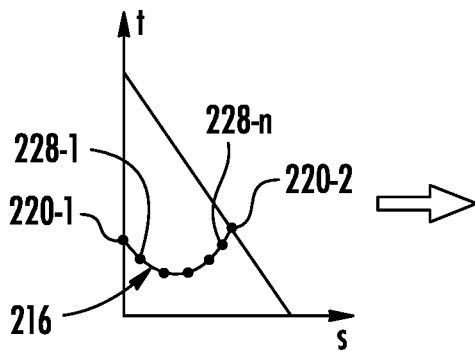


FIG. 6

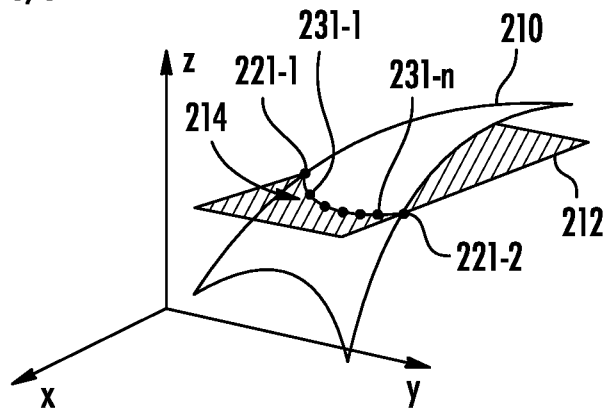
4/6



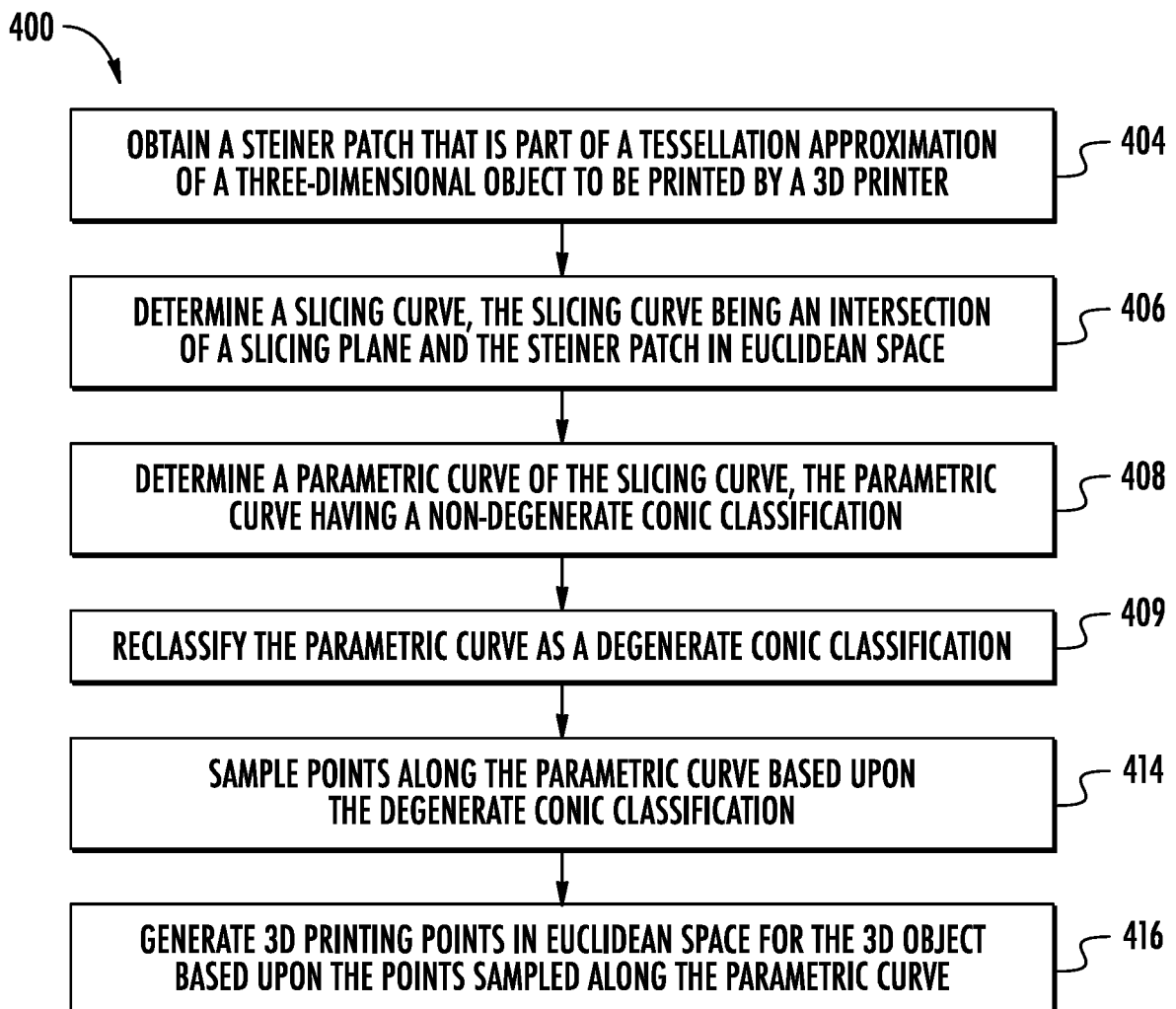
5/6



2D PARAMETRIC SPACE

**FIG. 10A**

3D EUCLIDEAN SPACE

**FIG. 10B****FIG. 11**



6/6

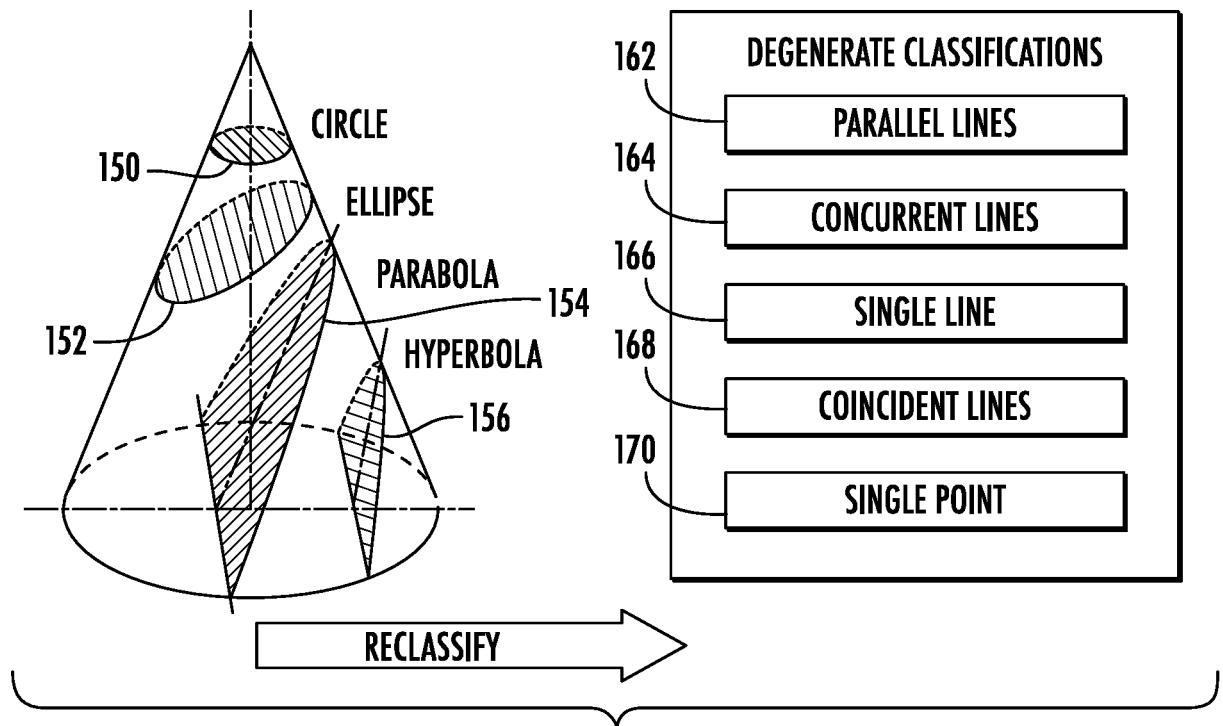


FIG. 12

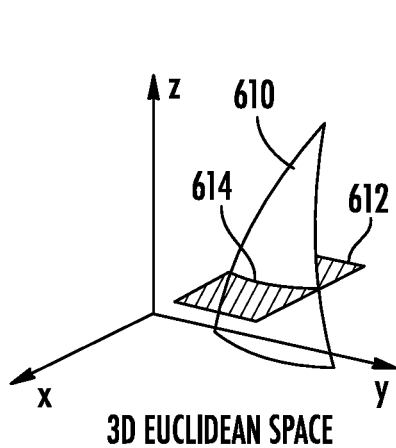


FIG. 13A

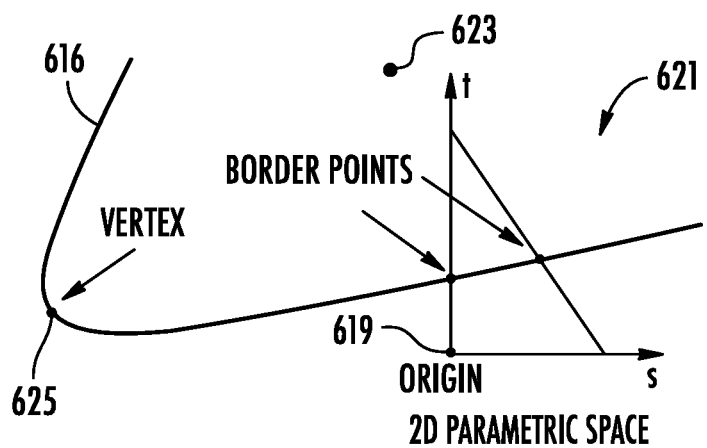


FIG. 13B

# INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 2020/012422

<b>A. CLASSIFICATION OF SUBJECT MATTER</b> <p style="text-align: center;"><i>G06T 17/20 (2006.01)</i></p> <p>According to International Patent Classification (IPC) or to both national classification and IPC</p>				
<b>B. FIELDS SEARCHED</b> <p>Minimum documentation searched (classification system followed by classification symbols)</p> <p style="text-align: center;">G06T 17/00, 17/20, 17/30, B29C 64/00, G06F 30/00</p> <p>Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched</p> <p>Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)</p> <p style="text-align: center;">PatSearch (RUPTO Internal), USPTO, PAJ, Espacenet, Information Retrieval System of FIPS</p>				
<b>C. DOCUMENTS CONSIDERED TO BE RELEVANT</b>				
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.		
A	US 7027050 B1 (CANON KABUSHIKI KAISHA) 11.04.2006, col. 1, lines 1-50	1-15		
A	US 9836884 B2 (DASSAULT SYSTEMES) 05.12.2017	1-15		
A	US 7113617 B2 (HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.) 26.09.2006	1-15		
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.				
<p>* Special categories of cited documents:</p> <table border="0"> <tr> <td style="vertical-align: top;"> <p>“A” document defining the general state of the art which is not considered to be of particular relevance</p> <p>“E” earlier document but published on or after the international filing date</p> <p>“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>“O” document referring to an oral disclosure, use, exhibition or other means</p> <p>“P” document published prior to the international filing date but later than the priority date claimed</p> </td> <td style="vertical-align: top;"> <p>“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>“&amp;” document member of the same patent family</p> </td> </tr> </table>			<p>“A” document defining the general state of the art which is not considered to be of particular relevance</p> <p>“E” earlier document but published on or after the international filing date</p> <p>“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>“O” document referring to an oral disclosure, use, exhibition or other means</p> <p>“P” document published prior to the international filing date but later than the priority date claimed</p>	<p>“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>“&amp;” document member of the same patent family</p>
<p>“A” document defining the general state of the art which is not considered to be of particular relevance</p> <p>“E” earlier document but published on or after the international filing date</p> <p>“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>“O” document referring to an oral disclosure, use, exhibition or other means</p> <p>“P” document published prior to the international filing date but later than the priority date claimed</p>	<p>“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>“&amp;” document member of the same patent family</p>			
Date of the actual completion of the international search <p style="text-align: center;">30 March 2020 (30.03.2020)</p>		Date of mailing of the international search report <p style="text-align: center;">14 May 2020 (14.05.2020)</p>		
Name and mailing address of the ISA/RU: Federal Institute of Industrial Property, Berezhkovskaya nab., 30-1, Moscow, G-59, GSP-3, Russia, 125993 Facsimile No: (8-495) 531-63-18, (8-499) 243-33-37		Authorized officer <p style="text-align: center;">V. Volodin</p> Telephone No. 8(495) 531-64-81		