

TESSELLATION OF A SURFACE BASED ON SLOPE

Background

The invention relates to tessellation of a surface, such as a non-uniform rational b-spline (NURBS) surface, for example.

Referring to Fig. 1, a graphics system may use a mathematical function to describe a
5 three dimensional (3-D) surface 5. For example, the graphics system may use a non-uniform
rational b-spline (NURBS) surface function to derive points of the surface 5. The NURBS
surface function is defined using elements called knots and control points. The control points
are located off of the surface 5 and are used to approximate the general curvature of the
surface 5. The knots are values that control which control points to use for a given region of
10 the surface 5.

Because the NURBS surface function is associated with a relatively small amount of
data, the NURBS representation of a surface may be well suited for purposes of transmitting a
representation of the surface 5 over a computer network having a low bandwidth, such as the
Internet, for example. This small data size may also be of interest to developers that create
15 highly detailed scenes.

Although the NURBS surface function is associated with a small amount of data, the
NURBS surface function may not be suited for purposes of rendering an image (of the surface
5) to be displayed by the system. Instead, for purposes of displaying an image of the surface
5, the graphics system may use a mesh of polygons to represent the surface 5, and the
20 graphics system may use the NURBS surface function to subdivide the surface 5 into the
polygons using a technique called tessellation. The end product of the tessellation typically is
a set of vertices (of the polygons) that describe the locations of the polygons. In this manner,
the graphics system may use the set of vertices to render an image of the surface 5.

In the course of the tessellation, the graphics system may select a vertex by estimating
25 an error between a polygon created with that vertex and the region of the surface that is
represented by the polygon. If the error is unacceptable, then the graphics system may resize
the polygon and repeat the error estimation. This process may continue until the polygon is
appropriately sized. During the tessellation, the graphics system may modify the original
NURBS surface function, such as by adding additional knots and control points to the
30 original NURBS surface function, as an example.

Unfortunately, the error estimation and the ongoing modification of the NURBS surface function may consume processing time of the system. As a result, the overhead incurred by the tessellation may limit the ability of the graphics system to rapidly display and manipulate images.

5 Thus, there is a continuing need for an arrangement that expedites the tessellation of mathematically described surfaces.

Summary

10 In one embodiment, a method for forming an image of a surface on a display of a computer system includes determining a first rate at which a slope of the surface changes near a first point of the surface and based on the determination, selecting a second point of the surface. At least one polygon is formed based on the first and second points to form at least a portion of the image. The polygon(s) are displayed on the display.

15 In another embodiment, an article includes a storage medium that is readable by a processor-based system. The storage medium stores instructions to cause a processor to determine a first rate at which a slope of a surface changes near a first point of the surface and based on the determination, select a second point of the surface. The storage medium stores instructions to cause the processor to tessellate at least a portion the surface based on the first and second points to form at least one polygon and cause the formation of said at least one
20 polygon on a display to represent at least a portion of an image of the surface.

 In yet another embodiment, a computer system includes a display, a processor and a memory. The memory stores a program to cause the processor to determine a first rate at which a slope of a surface changes near a first point of the surface and based on the determination, select a second point of the surface. The program further causes the processor
25 to tessellate at least a portion the surface based on the first and second points to form at least one polygon and cause the display to form said at least one polygon on the display to represent at least a portion of an image of the surface.

Brief Description Of The Drawing

30 Fig. 1 is a perspective view of a surface.

 Fig. 2 is a schematic diagram of a computer system according to an embodiment of the invention.

Fig. 3 is a perspective view of a surface illustrating tessellation.

Fig. 4 is an illustration of tessellation of surface by a processor of the computer system of Fig. 2.

5 Figs. 5, 6 and 7 are flow diagrams illustrating execution of a tessellation program by the processor of Fig. 2 according to an embodiment of the invention.

Detailed Description

Referring to Fig. 2, an embodiment 10 of a computer system in accordance with the invention includes a system memory 18 that stores a tessellation program 19. In some
10 embodiments, the tessellation program 19, when executed by a processor 12 (of the computer system 10), causes the processor 12 to tessellate a mathematically described surface 40 (see Fig. 3) to form a mesh of polygons that represents the surface 40. In this manner, the processor 12 selects a set of vertices of the polygons, and these vertices may be used for purposes of quickly rendering an image of the surface 40 on a display 14 of the computer
15 system 10. As an example, the surface 40 may be a non-uniform, rational b-spline surface (NURBS).

In the course of selecting each vertex, the processor 12 (when executing the program 19) takes into account the rate at which the slope of the surface 40 changes near that vertex. As described below, as a result of this technique, the processor 12 represents regions 42 (see
20 Fig. 3) in which the slope of the surface 40 changes rapidly with smaller polygons and represents regions 43 in which the slope of the surface 40 remains nearly constant with larger polygons, as described below. In some embodiments, the processor 12 performs these functions without manipulating the mathematical function that describes the surface 40 and without estimating subdivision errors, techniques that typically consume processing power.

25 Thus, the advantages of the above-described arrangement may include one or more of the following: tessellation processing times may be reduced; a mathematical function that describes the surface may not be altered; single-instruction-multiple-data (SIMD) instructions may be used; surfaces may be tessellated in real time; and over-tessellation may be avoided.

In the context of this application, the phrase "computer system" may generally refer to
30 a system that includes a processor and may include (but is not limited to) a graphics system, a desktop computer or a mobile computer (a laptop computer, for example), as just a few examples. The term "processor" may refer to, as examples, to at least one microcontroller,

- 4 -

X86 microprocessor, Advanced RISC Machine (ARM) microprocessor or Pentium microprocessor. The examples given above are not intended to be limiting, but rather, other types of computer systems and other types of processors may be included in embodiments of the invention.

- 5 Referring to Fig. 3, in some embodiments, the surface 40 may be described by a NURBS surface function (called $S(u,v)$) that is obtained by mapping a bounded two-dimensional parameter space (u,v) into a set of points that describes the surface 40. The $S(u,v)$ surface function is described below:

$$S(u,v) = \frac{\sum_{i=0}^p \sum_{j=0}^q B_{i,p}(u) B_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^p \sum_{j=0}^q B_{i,p}(u) B_{j,q}(v) w_{i,j}},$$

- 10 where P_{ij} represents an $m \times n$ array of control points that control the curvature of the surface 40; w_{ij} are control point weights; and $B_{i,p}(u)$ and $B_{j,q}(v)$ are b-spline basis functions defined by the Cox-DeBoor recurrence formula over the following knot vectors called U and V :

$$U = \{u_0, u_1, u_2, u_p, \dots, u_{m+p-2}, u_{m+p-1}, u_{m+p}\} \text{ and}$$

$$V = \{v_0, v_1, v_2, v_q, \dots, v_{n+q-2}, v_{n+q-1}, v_{n+q}\}$$

- 15 The B-spline basis functions are described below:

$$B_{i,0}(u) = 1, \text{ if } u_1 \leq u < u_{i+1}$$

$$B_{i,0}(u) = 0, \text{ otherwise}$$

$$B_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} B_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} B_{i+1,p-1}(u)$$

The $S(u,v)$ surface function is defined to be valid over the closed knot range from $(u_{\min}, v_{\min})$ to $(u_{\max}, v_{\max})$. The x, y, z and homogenous w components of the $S(u,v)$ surface function, denoted by S_x, S_y, S_z and S_w , respectively, are described by the following equations:

20
$$S_x(u,v) = \sum_{i=0}^p \sum_{j=0}^q B_{i,p}(u) B_{j,p}(v) w_{i,j} P_{x,i,j}$$

$$S_y(u,v) = \sum_{i=0}^p \sum_{j=0}^q B_{i,p}(u) B_{j,q}(v) w_{i,j} P_{y,i,j}$$

$$S_z(u,v) = \sum_{i=0}^p \sum_{j=0}^q B_{i,p}(u) B_{j,q}(v) w_{i,j} P_{z,i,j}$$

$$S_w(u, v) = \sum_{i=0}^p \sum_{j=0}^q B_{i,p}(u) B_{j,q}(v) w_{i,j}$$

In the following description, partial derivatives are designated using superscripts. For example:

$$5 \quad \frac{\partial S(u, v)}{\partial u} \equiv S^u \text{ and } \frac{\partial}{\partial v} \left(\frac{\partial S(u, v)}{\partial v} \right) \equiv S^{vv}$$

Referring to Fig. 4, in some embodiments, the polygons may be triangles 44, and the processor 12 may select the vertices of the triangles 44 based on first and second partial derivatives of the $S(u, v)$ surface function. For example, the processor 12 may select a particular vertex 46a and then, using the $S(u, v)$ surface function, calculate the rate of change of the slope of the surface 40 at the vertex 46a with respect to either the parameter u or v . For example, the processor 12 may hold the u parameter constant and advance along a v -parametric curve 48 to select the next vertex 46b (after the vertex 46a) by incrementing the value of the v parameter. In this manner, the processor 12 may calculate the S^{vv} second partial derivative at the vertex 46a and from this calculation, determine an amount by which to increase the v parameter. After the processor 12 increases the v parameter by this amount, the processor 12 uses the $S(u, v)$ surface function (with the new v parameter) to determine the next vertex 48b.

Similarly, in the course of determining the vertices, the processor 12 may hold the v parameter constant and advance along the resulting u -parametric curve by changing the u parameter in manner similar to the technique described above. In this manner, the processor 12 may calculate the S^{uu} partial derivative at each vertex and advance along the v -parametric curve based on the calculation.

In general, if the second partial derivative (S^{vv} or S^{uu}) is approximately equal to zero, the processor 12 may advance the parameter (u or v) by a large value to set a relatively large distance between adjacent vertices, as the slope of the surface 40 changes by a relatively small amount between the vertices. However, if the second partial derivative is changing by a significant amount, then the processor 12 may advance the parameter by a small value to minimize the distance between adjacent vertices, as the slope of the surface 40 is changing more rapidly. The processor 12 may also advance the parameter by a small amount in the

neighborhood of curvature discontinuities (as indicated by the first and second partial derivatives of the $S(u,v)$ surface function), as further described below.

In some embodiments, without curvature discontinuities, the processor 12 generally sets the advancement of the u or v parameter to be inversely proportional to the absolute value of the partial second derivative of the $S(u,v)$ surface function with respect to that parameter. However, other arrangements are possible. For example, the parameter may be advanced according to a schedule of values. In this manner, the larger values of the schedule may be associated with second partial derivatives having smaller magnitudes, and the smaller values of the schedule may be associated with second partial derivatives having larger magnitudes.

As an example, the program 19 may cause the processor 12 to select vertices of the polygon mesh in the following manner. Starting with the minimum boundary value for the u parameter (called $u_{\min}$), the processor 12 advances the v parameter from $v_{\min}$ (the minimum boundary value for the v parameter) to $v_{\max}$ (the maximum boundary value for the v parameter) along a v -parametric curve. Each advancement of the v parameter produces a selected vertex, and the processor 12 advances to the next selected vertex based on the value of the S'' partial derivative at the last selected vertex. The processor 12 then advances the u parameter based on the S''' partial derivative and repeats the above-described process to select more vertices. When the processor 12 determines that the u parameter has reached $u_{\max}$ (the maximum boundary value for the u parameter), the selection of the vertices (and the tessellation of the surface 40) is complete.

Referring to Fig. 5, more particularly, upon execution of the program 19, the processor 12 may initially set the u (block 50) and v (block 52) parameters to their minimum $u_{\min}$ and $v_{\min}$ boundary values. Next, the processor 12 may determine (diamond 54) whether the u parameter is less than or equal to $u_{\max}$. If not, the tessellation is complete, and the processor 12 may terminate execution of the program 19. Otherwise, the processor 12 determines (diamond 56) whether the v parameter is less than or equal to $v_{\max}$, i.e., the processor 12 determines whether it is time to advance the u parameter.

If the v parameter is less than or equal to $v_{\min}$, the processor 12 determines (block 58) the first (S') and second (S'') partial derivatives at the current point (u,v) and adds (block 60) the current point (given by the $S_x(u,v)$, $S_y(u,v)$ and $S_z(u,v)$ functions) to the set of selected vertices.

Next, the processor 12 determines an amount (called v_{DELTA}) in which to increase the v parameter. Referring to Fig. 6, in this manner, if the processor 12 determines (diamond 62) that the second partial derivative (S'') is approximately equal to zero, then the processor 12 subsequently determines (diamond 66) whether the first partial derivative (S') is approximately equal to zero. If both the first (S') and second (S'') partial derivatives are equal to zero, then the processor 12 may assume that the surface 40 is substantially planar with respect to the v parameter near the current point given by the u and v parameters. If this is the case, then the processor 12 may set v_{DELTA} to a relatively large value. For example, the processor 12 may advance v_{DELTA} proportionally to the difference between the two NURBS knots that are located on both sides of the v parameter. In this manner, the processor 12 may set (block 70) v_{DELTA} equal to:

$$\frac{v_{\text{KNOT}+1} - v_{\text{KNOT}}}{q},$$

where $v_{\text{KNOT}} \leq v \leq v_{\text{KNOT}+1}$.

If the processor 12 determines (diamond 66) that the first partial derivative (S') is not approximately equal to zero (but the second partial derivative S'' is approximately equal to zero as determined in diamond 62), a condition that indicates a discontinuity in the curvature of the surface 40, then the processor 12 sets v_{DELTA} to a small predetermined value (called EPSILON). If the processor 12 determines (diamond 62) that the S'' second partial derivative is not approximately equal to zero, then the processor 12 sets (block 64) v_{DELTA} to be inversely proportional to the absolute value of the S'' second partial derivative.

After the processor 12 determines the value for v_{DELTA} , the processor 12 sets (block 72) the new value for the v parameter equal to the summation of the current value for the v parameter and v_{DELTA} . However, if the processor 12 determines (diamond 74) that the new value of the v parameter is greater than the value of the $v_{\text{KNOT}+1}$ knot value, then the processor 12 sets (block 76) the new value equal to the $v_{\text{KNOT}+1}$ value.

Referring back to Fig. 5, the processor 12 subsequently determines (diamond 56) if the value for the v parameter exceeds v_{MAX} . If not, the processor 12 continues to perform the functions associated with the constant u , as described above. However, if so, the processor 12 determines an amount (called u_{DELTA}) in which to increase the value of the u parameter. In this manner, referring to Fig. 7, the processor 12 determines (block 77) the S'' and S''' partial derivatives. In some embodiments, in this determination, the processor 12 may evaluate the

S'' and S''' partial derivatives at the last selected value of v along the last v parametric curve, as an example. However, other selected values of v may be used for this evaluation. In some embodiments, the processor 12 may evaluate the S'' and S''' partial derivatives for each selected value of v along the last v parametric curve and choose the smallest u advancement based on the results, as further described below; however, for purposes of describing the flow illustrated in Fig. 7, it is assumed that the processor 12 evaluates one S'' partial derivative and one S''' partial derivative at a selected v value to determine the u advancement.

If the processor 12 determines (diamond 78) that the S''' second partial derivative is approximately equal to zero, then the processor 12 subsequently determines (diamond 82) whether the S'' first partial derivative is approximately equal to zero. If both the first (S'') and second (S''') partial derivatives are equal to zero, then the processor 12 may assume that the surface 40 is substantially planar with respect to the u parameter near the current point given by the u and v parameters. If this is the case, then the processor 12 may set u_{DELTA} to a relatively large value. For example, the processor 12 may set u_{DELTA} to proportionally the difference between the two knots that are located on both sides of u . In this manner, the processor 12 may set (block 86) u_{DELTA} equal to:

$$\frac{u_{\text{KNOT}+1} - u_{\text{KNOT}}}{P},$$

where $u_{\text{KNOT}} \leq u \leq u_{\text{KNOT}+1}$.

If the processor 12 determines (diamond 82) that the S'' first partial derivative is not approximately equal to zero (but the S''' second partial derivative is approximately equal to zero), a condition that indicates a discontinuity in the curvature of the surface 40, then the processor 12 sets u_{DELTA} to EPSILON. If the processor 12 determines (diamond 78) that the S''' second partial derivative is not approximately equal to zero, then the processor 12 sets (block 80) u_{DELTA} to be inversely proportional to the absolute value of the S''' second partial derivative.

After the processor 12 determines the new u_{DELTA} , the processor 12 then sets (block 88) the new value for the u parameter equal to the summation of the current value for the u parameter and u_{DELTA} . However, if the processor 12 determines (diamond 90) that the new value for the u parameter is greater than the value of the $u_{\text{KNOT}+1}$ knot value, then the processor 12 sets (block 92) the new value equal to the $u_{\text{KNOT}+1}$ value. The processor 12 subsequently returns to diamond 54, as depicted in Fig. 5.

Referring back to Fig. 2, in some embodiments, the computer system 10 may include a bridge, or memory hub 16. The processor 12 and the memory hub 16 may be coupled to a host bus 14. The memory hub 16 may provide interfaces to couple the host bus 14, a memory bus 29 and an Accelerated Graphics Port (AGP) bus 11 together. The AGP is described in
5 detail in the Accelerated Graphics Port Interface Specification, Revision 1.0, published on July 31, 1996, by Intel Corporation of Santa Clara, California. The system memory 18 may be coupled to the memory bus 29, and a display controller 13 (that controls the display 14) may be coupled to the AGP bus 11. A hub communication link 15 may couple the memory hub 16 to another bridge circuit, or input/output (I/O) hub 20.

10 In some embodiments, the I/O hub 20 includes interfaces to an I/O expansion bus 25 and a Peripheral Component Interconnect (PCI) bus 21. The PCI Specification is available from The PCI Special Interest Group, Portland, Oregon 97214. The I/O hub 20 may also include interfaces to a hard disk drive 32 and a CD-ROM drive 33, as examples. An I/O controller 17 may be coupled to the I/O expansion bus 25 and receive input data from a
15 keyboard 24 and a mouse 26, as examples. The I/O controller 17 may also control operations of a floppy disk drive 22. Copies of the program 19 may be stored on, as examples, the hard disk drive 32, a diskette or a CD-ROM, as just a few examples.

Other embodiments are within the scope of the following claims. For example, in the execution of the above-described tessellation program 19, the processor 12 determines
20 advances for the u and v parameters separately by evaluating the S^v and S^{vv} partial derivatives along each v parametric curve to determine the v advancements and subsequently determining the S^u and S^{uu} partial derivatives to determine the u advancement before advancing along the next v parametric curve. However, alternatively, the processor 12 may determine the parametric advancements by evaluating the S^{vv} , S^v , S^u , and S^{uu} partial
25 derivatives together.

For example, in some embodiments, in the "v-parameter loop" (i.e., blocks and diamonds 58-76 of Figs. 5 and 6), the processor 12 may calculate the S^{uu} and S^u partial derivatives for each selected value of v along the particular v parametric curve, and from these partial derivatives, the processor 12 may determine different amounts in which to
30 advance the u parameter for the next v -parameter loop. In this manner, when the processor 12 completes the advancement along the particular v parametric curve, the processor 12 may

- 10 -

choose the smallest (as an example) advancement (from the calculated u advancements) by which to advance the u parameter before advancing along the next v parametric curve.

As other examples, the above-described techniques may be applied to select points of the polygons other than vertices of the polygons. The above-described techniques may also
5 be applied to the tessellation of volumes. The polygons may be polygons other than triangles.

While the invention has been disclosed with respect to a limited number of embodiments, those skilled in the art, having the benefit of this disclosure, will appreciate numerous modifications and variations therefrom. It is intended that the appended claims
10 cover all such modifications and variations as fall within the true spirit and scope of the invention.

What is claimed is:



- 1 1. A method for forming an image of a surface on a display of a computer
2 system, comprising:
3 determining a first rate at which a slope of the surface changes near a first
4 point of the surface;
5 based on the determination, selecting a second point of the surface;
6 forming at least one polygon based on the first and second points to form at
7 least a portion of the image; and
8 displaying said at least one polygon on the display.
- 1 2. The method of claim 1, wherein the act of determining comprises:
2 determining a partial derivative of a mathematical function that describes the
3 surface.
- 1 3. The method of claim 2, wherein the mathematical function comprises:
2 a non-uniform rational b-spline surface function.
- 1 4. The method of claim 1, wherein the act of selecting comprises:
2 advancing a distance from the first point to the second point; and
3 regulating the distance based on the first rate.
- 1 5. The method of claim 4, wherein the act of regulating comprises:
2 downwardly adjusting the distance when the first rate has a larger value; and
3 upwardly adjusting the distance when the first rate has a smaller value.
- 1 6. The method of claim 4, wherein the act of regulating comprises:
2 causing the distance to be inversely proportional to the first rate.
- 1 7. The method of claim 1, wherein the act of forming said at least one polygon
2 comprises:
3 using the first and second points to form vertices of the polygon.



1 8. The method of claim 1, further comprising:
2 determining a second rate at which a slope of the surface changes near the
3 second point of the surface;
4 based on the determination, selecting a third point of the surface; and
5 using the third point in the formation of said at least one polygon.

1 9. The method of claim 1, further comprising:
2 determining other rates at which slopes of the surface change near other points
3 of the surface;
4 using the other rates to select additional points of the surface; and
5 using the additional parts to form additional polygons.

1 10. An article comprising a storage medium readable by a processor-based system,
2 the storage medium storing instructions to cause a processor to:
3 determine a first rate at which a slope of a surface changes near a first point of
4 the surface,
5 based on the determination, select a second point of the surface,
6 tessellate at least a portion the surface based on the first and second points to
7 form at least one polygon, and
8 cause the formation of said at least one polygon on a display to represent at
9 least a portion of an image of the surface.

1 11. The article of claim 10, the storage medium storing instructions to cause the
2 processor to determine a partial derivative of a mathematical function that describes the
3 surface.

1 12. The article of claim 11, wherein the mathematical function comprises:
2 a non-uniform rational b-spline surface function.



1 13. The article of claim 10, the storage medium storing instructions to cause the
2 processor to:
3 advance a distance from the first point to the second point, and
4 regulate the distance based on the first rate.

1 14. The article of claim 13, the storage medium storing instructions to cause the
2 processor to:
3 downwardly adjust the distance when the first rate has a larger value, and
4 upwardly adjust the distance when the first rate has a smaller value.

1 15. The article of claim 13, the storage medium storing instructions to cause the
2 processor to:
3 cause the distance to be inversely proportional to the first rate.

1 16. The article of claim 10, the storage medium storing instructions to cause the
2 processor to:
3 use the first and second points to form vertices of the polygon.

1 17. The article of claim 10, the storage medium storing instructions to cause the
2 processor to:
3 determine a second rate at which a slope of the surface changes near the
4 second point of the surface,
5 based on the determination, select a third point of the surface, and
6 use the third point in the formation of said at least one polygon.

1 18. The article of claim 10, the storage medium storing instructions to cause the
2 processor to:
3 determine other rates at which slopes of the surface change near other points of
4 the surface,
5 use the other rates to select additional points of the surface; and
6 use the additional points to form additional polygons.



1 19. A computer system comprising:
2 a display;
3 a processor; and
4 a memory storing a program to cause the processor to:
5 determine a first rate at which a slope of a surface changes near a first point of
6 the surface,
7 based on the determination, select a second point of the surface,
8 tessellate at least a portion the surface based on the first and second points to
9 form at least one polygon, and
10 cause the display to form said at least one polygon on the display to represent
11 at least a portion of an image of the surface.

1 20. The computer system of claim 19, wherein the processor is adapted to
2 determine the first rate by determining a partial derivative of a mathematical function that
3 describes the surface.