Free Form Surface Construction Using Gregory Patches

Shouqing Zhang, A. L. Thomas
Spatial Model Animation and Machine Vision Research Group
School of Engineering, The University of Sussex
Falmer, Brighton BN1 9QT, U.K.
e-mail: S.Zhang@central.sussex.ac.uk

ABSTRACT

This paper deals with the construction of a free form surface by using both triangular and rectangular Gregory patches. The input is a set of $3D$ points associated with normal values at each point. All the control points needed for a Gregory patch can then be calculated automatically to achieve the G^1 continuity between neighbouring patches within surface.

Key Words: *Gregory Patch, Triangular and Rectangular shape, G^1 Continuity*

1 Introduction.

The importance of free-form surface representation in the field of CAD/CAM is well established, but a considerable amount of research has been necessary to establish a pratical method. The Chiyokura and Kimura's method [2] supplies an elegant and yet still easy to use method. The "Gregory patch", which was what Chiyokura and Kimura named this method, could be thought of as an extension of Bezier's rectangular bicubic patch. Instead of having four inner control points for each patch as for the ordinary Bezier patch, Chiyokura and Kimura used a modified version of Gregory's squares requiring eight inner control points[3]. For the triangular Gregory patch, a similar method is employed to set up six inner control points instead of three to utilise the triangular quartic Bezier surface patch. The cross-derivative vectors at the corners of the patches can be specified independently. In this method, the inner control points are used to decouple the effect of constraints of control points when the connection of neighbouring patches is considered. In both publications of [2] and [3], this surface scheme was used in an interactive environment to build a desired shape. In these schemes normals were not made use of, and the connection between triangular and rectangular Gregory patches was not reported. In this paper, a method is proposed to set up the boundary curve model by a very simple vector manipulation operation taking advantage of the normals associated with each vertex. The desired shape is then achieved if a set of mixed rectangularized and triangularized data point, together with their associated normals are properly specified. This boundary curve network is then used to produce the necessary inner control points, and finally, to generate the surface equations interpolating over the curve network .

2 Free-Form Shape Design Process.

It is not difficult to set up all the control points for a single isolated Gregory or Bezier rectangular patch since the specification of control points is only dependant on the shape desired. For a smoothly connected multi-patch surface, i.e, the so called free-form surface, the specification of control points for each individual patch is no longer independant. The continuation across the common boundaries of any two neighbouring patches requires that these relevant control points be set up according to some rules.

Suppose we have a collection of a mixture of rectangularized and triangularized $3D$ data points, and a normal is associated with each point. A variety of interpolation schemes can de developed from this framework. For the Gregory patch method, the construction consists of the following three steps,

- First, define a polyhedral model which has triangular and rectangular faces to serve as the basis for the free-form shape,
- Second, automatically generate a curve model for the free-form shape from this polyhedral model,
- Finally, generate surface equations interpolating over the curve model.

[2]

A polyhedral model can be achieved by data sampling and then by using some rectangularization (or triangulation if necessary) method, to set up the topological relationships among the $3D$ data points. There are then several ways of establishing the curve model in the second step. Here, a method based on vector manipulation is proposed using the normal associated with each point. The surface patch equations are then produced by using the Gregory patch method.

A brief description of the second and third steps will be given in the following sections.

2.1 Set up the curve model.

In some applications, the normals may not be supplied with the surface data points. However, there are several ways of setting up the required normal network. The choice of the normal specification helps to determine the required shape.

Let $N_{ijk}, i,j = 0,3, k = 0$, denotes the normal at points $\mathbf{b}_{ijk}$, $i,j = 0,3, k = 0$, which are four distinct points in $3D$ space forming rectangular shape, shown on Figure 1.

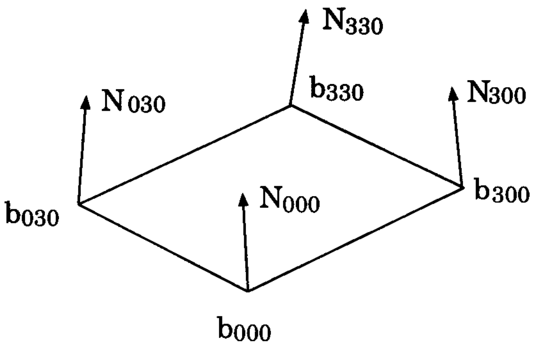

Figure 1: A $3D$ rectangle with normals at the four corners.

A tangent plane can be set up at each of these four corner points:

$$N_{ijk} \cdot (\mathbf{b} - \mathbf{b}_{ijk}) = 0$$

where, $i,j = 0,3, k = 0$ and $\mathbf{b}$ is any point on this tangent plane.

As can be seen from Figure 1, each edge is common to two neighbouring rectangles. The control points along each edge should be shared by these two rectangles so as to ensure G^0 continuity between them.

In order to ensure G^1 continuity, in another word, tangent plane continuity, along the commmon boundary curves, the control points on both patches must all agree with the cross-boundary tangent fields [2]. As a result of this construction, all control points surrounding each corner point must lie on the same tangent plane passing through the corner point. The following paragraphs will show how to employ this fact to achieve the curve model.

It is assumed that the normals at each point are unique to the corresponding vertex points. Thus, the cross product of N_{ijk} and N_e (which is an edge vector of the rectangle where $N_e = \mathbf{b}_{030} - \mathbf{b}_{000}$, current control point is $\mathbf{b}_{000}$) will form another vector,

$$N_c = N_{ijk} \times N_e$$

which is perpendicular to the plane formed by N_{ijk} and N_e. Taking another cross product of vectors N_c and N_{ijk}, we have

$$N_d = N_c \times N_{ijk}$$

N_d is the intersection vector of the tangent plane at point $\mathbf{b}_{ijk}$ with the plane formed by vectors N_{ijk} and N_e. This vector points to the other end of the current edge and ensures that the calculated control point corresponds to that edge. Control points surrounding $\mathbf{b}_{ijk}$ can be set up along all these emanating vectors, while the distances from $\mathbf{b}_{ijk}$ can be any scaled length of the corresponding edges. Once all the points in the collection have been processed, a curve network over this data collection will have been set up in terms of the necessary control points for each edge cubic curve (refer to Figure 2). These control points will be indexed by subscripts like $\mathbf{b}_{ijk}, i, j = 0, 1, 2, 3, k = 0$.

If the two seperate steps are combined into one formula, we can use the representation,

$$N_d = N_{ijk} \times N_e \times N_{ijk}$$

This procedure will be carried out along each of the edges which are emanating from the current vertex $\mathbf{b}_{ijk}$, and a measure of distance along each of these calculated vectors will be taken as a scaled length of the corresponding edge to get the desired new control points, $\mathbf{b}_i, i = 1...4$ as shown on Figure 2.

Thus, the initial curve network will guarantee that all the surface patches sharing this corner point will share the same tangent plane.

This procedure makes the calculation very efficient and time saving. If a suitable data structure is combined with this method (such as a wing-edged structure), all the control points, edge and surface information is easy to handle. The calculated control points (for the curve network) can be stored in this structure for further processing. To save memory, these vertex data points together with the inner control points can be discarded after each surface patch has been build up. They can be calculated later if needed. This calculation does not take much time.

The calculation for triangular patches is similar except for the index of the control points. As a result, there will be six inner control points for such Gregory triangular patch instead of three for ordinary quartic Bézier patch, [4], as is shown in Figure 3.

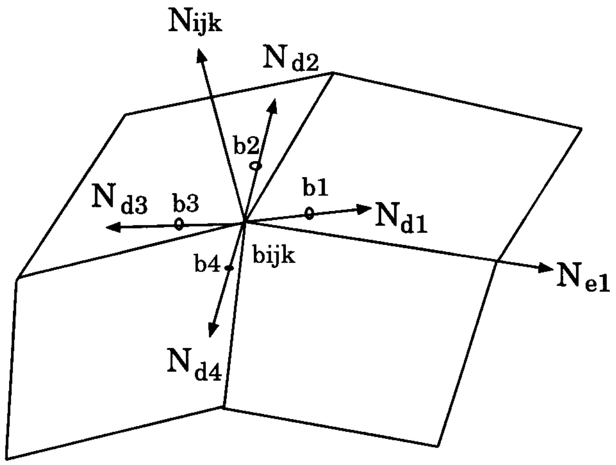

Figure 2: Emanating vectors from current vertex and the corrsponding control points.

2.2 Surface Interpolation.

From the calculated curve network, we come to the next stage which is to calculate the inner control points for each patch.

For rectangular patches, Chiyokura and Kimura used a "basis patch" to solve the eight inner control points problem while satisfying the continuity condition along the common boundary curves. [2]. The coupling control points will be distinguished by subscript $k = 0$, or $k = 1$.

The final interpolation surface equation is similar to the bicubic Bezier surface representation except that the control points are handled in different way,

$$\mathbf{b}^{3,3}(u,v) = \sum_{i=1}^{3}\sum_{j=1}^{3} \mathbf{p}_{ij} B_i^3(u) B_j^3(v), \qquad 0 \le u, v \le 1$$

As can be seen from Figure 4, we have calculated all together 20 control points, while in the above surface equation, only 16 control points are needed. The eight inner points are blended into four in the following way:

$\mathbf{p}_{ij}(u,v) = \mathbf{b}_{ij0} = \mathbf{b}_{ij1}$, except for $\mathbf{p}_{11}, \mathbf{p}_{21}, \mathbf{p}_{12}, \mathbf{p}_{22}$, and we have

$$\mathbf{p}_{11}(u,v) = \frac{u\mathbf{b}_{110} + v\mathbf{b}_{111}}{u+v}$$

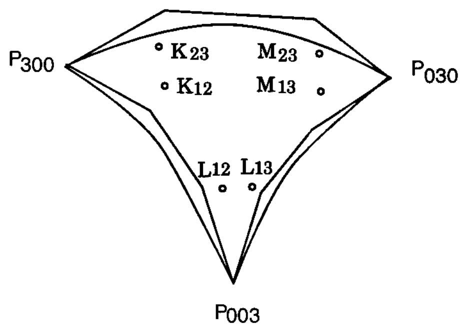

Figure 3: Control points of a triangular Gregory patch.

$$\mathbf{p}_{21}(u,v) = \frac{(1-u)\mathbf{b}_{210} + v\mathbf{b}_{211}}{1-u+v}$$

$$\mathbf{p}_{12}(u,v) = \frac{(u)\mathbf{b}_{120} + (1-v)\mathbf{b}_{121}}{u+1-v}$$

$$\mathbf{p}_{22}(u,v) = \frac{(1-u)\mathbf{b}_{220} + (1-v)\mathbf{b}_{211}}{2-u-v}$$

Each rectangular patch will be processed in this way, and the required G^1 continuity along all the common boundaries will be achieved.

For triangular patches, the Gregory scheme uses the method of Chiyokura and Kimura to find a pair of cross-boundary points for each edge (Figure 3) [1] and [4]. In this figure, each of the three point pairs, L_{12} and L_{13}, K_{12} and K_{23}, and M_{23} and M_{13}, will be blended to form a single point when evaluating the patch at a domain point (u,v,w), and thus the six inner control points will be blended into three. This yields a quartic Bézier Patch.

Here, each point pair will be blended according to the following formula (take L_{12} and L_{13} for example):

$$\mathbf{L}_{112}(u,v,w) = \frac{w(1-v)L_{13} + v(1-w)L_{12}}{w(1-v)+v(1-w)}$$

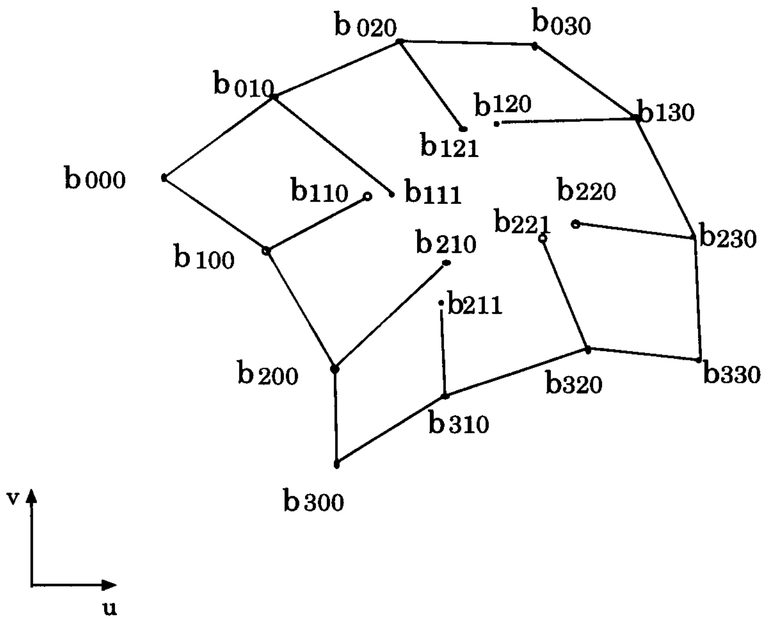

Figure 4: Control point layout for a rectangular Gregory patch.

We have cubic curve boundaries for each triangle. In order to make use of the quartic Bézier patch, these cubic boundaries must be degree elevated to quartic ones to match the three blended inner control points. Finally, the quartic Bézier patch can be applied to interpolate the control point set.

2.3 Application

As an example, a set of $3D$ points are sampled from a vase-like shape. The Body of the vase is formed by rectangles while the "cap" of the vase consistes of a dozen triangles. The normals for each point are averaged from the normal sum of the surrounding shapes, as is shown in Figure 5.

Next, using the methods mentioned above, all the control points for each patch are obtained and the final surface is filled in. Figure 6 shows the result.

2.4 Discussion

Although a so called free-form surface can be built up in this way, problems still exist. Gregory patches supply us a simple way of dealing with the cross-boundary continuity problem, being

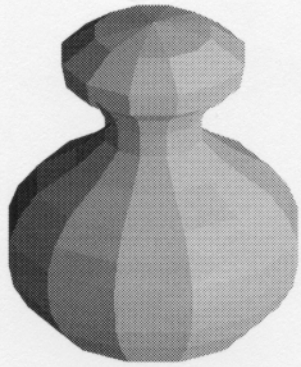

Figure 5: Framework for a vase-like shape.

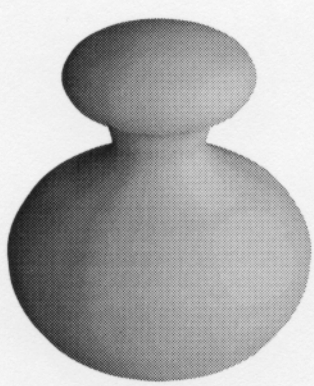

Figure 6: Smooth connection between different patches using Gregory method.

very efficient in terms of computation and display compared with other split-domain schemes, but it is still difficult to guarantee surface quality when the surface shape becomes complicated.

References

[1] S. Mann et al. A survey of parametric scattered data fitting using triangular interpolants. In H. Hagen, editor, *Curves and Surface Design.* SIAM,Philadelphia, 1992.

[2] H.Chiyokura. Localized surface interpolation method for irregular meshes. In Tosiyasu L. Kunii, editor, *Advanced Computer Graphics.* Springer-Verlag, 1986.

[3] H.Chiyokura and F.Kimura. Design of solids with free-form surfaces. *Computer Graphics*, 17:289–298, 1983.

[4] L.Longhi. Interpolating patches between cubic boundaries. Technical Report 87/313, University of California, Berkeley, 1986.