Geometric Modeling Notes

CATMULL-CLARK SURFACES

Kenneth I. Joy
Institute for Data Analysis and Visualization
Department of Computer Science
University of California, Davis

Overview

Utilizing the subdivision for bicubic uniform B-spline surfaces, Ed Catmull and Jim Clark, following the methodology of Doo and Sabin noted that the subdivision rules expressed for the cubic B-spline surface not only work for arbitrary rectangular meshes, but can also be extended to meshes of an arbitrary topology. This extension was accomplished by generalizing the definition of a face point, by modifying the method for calculating vertex points (which extends that of the uniform B-spline case), and by specifying a method for reconnecting the points into a mesh.

Specifying the Refinement Procedure

Given a mesh of control points with an arbitrary topology, we can generalize the face point, edge point, vertex point specification from the uniform B-spline surface calculations to obtain

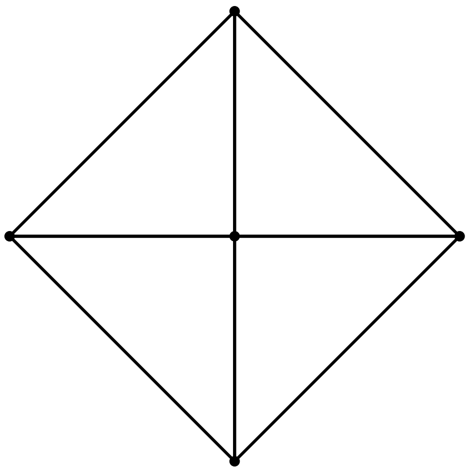
- For each face of the mesh, generate the new face points – which are the average of all the original points defining the face (We note that faces may have 3, 4, 5, or many points now defining them).
- Generate the new edge points – which are calculated as the average of the midpoints of the original edge with the two new face points of the faces adjacent to the edge.
- Calculate the new vertex points – which are calculated as the average of $\mathbf{Q}$, $2\mathbf{R}$ and $\frac{(n-3)\mathbf{S}}{n}$, where $\mathbf{Q}$ is the average of the new face points of all faces adjacent to the original face point, $\mathbf{R}$ is the average of the midpoints of all original edges incident on the original vertex point, and $\mathbf{S}$ is the original vertex point.

The mesh is reconnected by the following method.

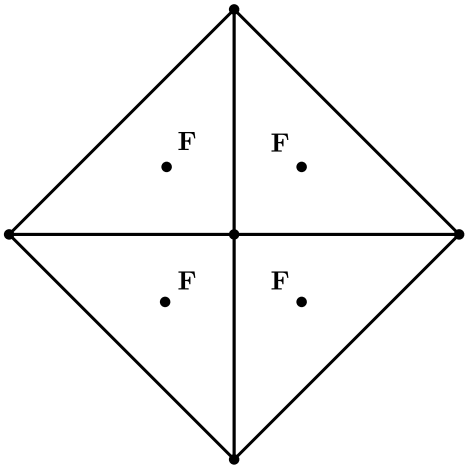
- Each new face point is connected to the new edge points of the edges defining the original face.
- Each new vertex point is connected to the new edge points of all original edges incident on the original vertex point.

Example

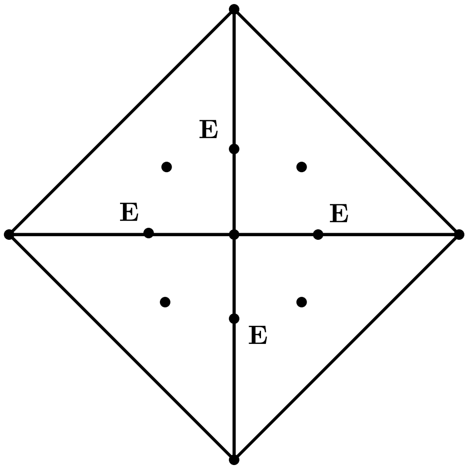
For an example of this process consider the mesh consisting of four triangles in a diamond pattern, as is illustrated below.



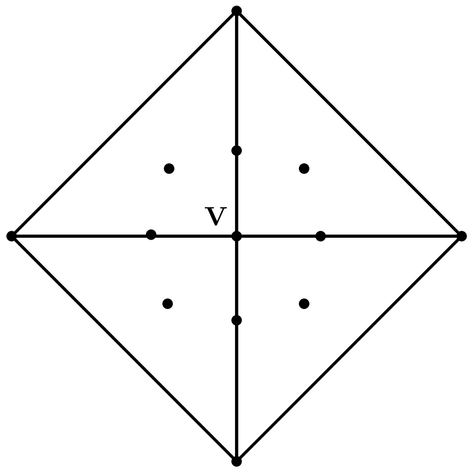
First, we construct the face points, which are calculated as the average of the points that make up each of the original faces. These points are shown in the figure below, labeled with an $\mathbf{F}$.



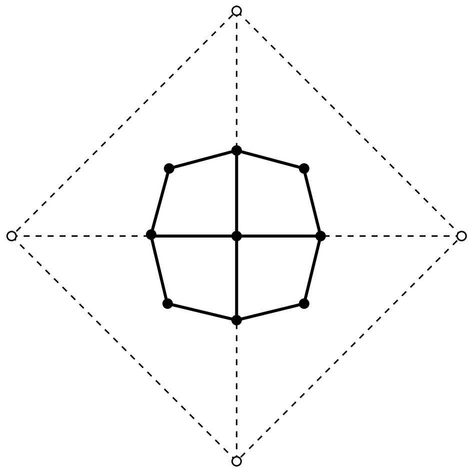
Now, we construct the edge points, which are calculated as the average of the four points – the two original points which define the edge, and the two new face points for the faces that are adjacent to the edge. These points are shown in the figure below and are labeled with an $\mathbf{E}$.



We now construct the single vertex point. This point, at least in two dimensions, is identical with the center of the diamond. This point is shown in the figure below.

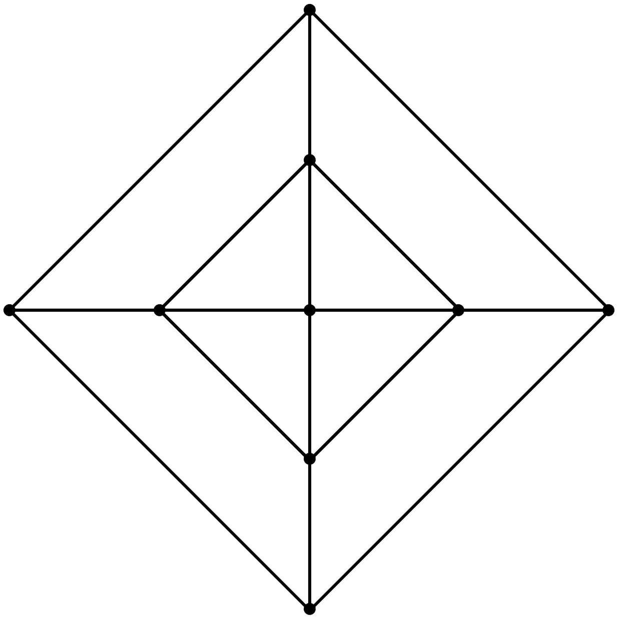


Finally, we connect the edges to the points we have generated: First connecting the face points to the edge points that correspond to edges on the face, and then connecting the vertex point to the edge points.

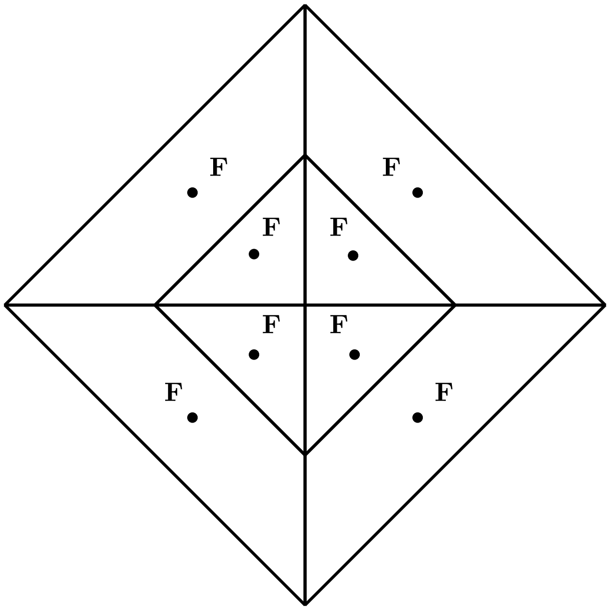


We note now that the each face of the refined mesh has four edges, and our above algorithm with four edges can now be used.

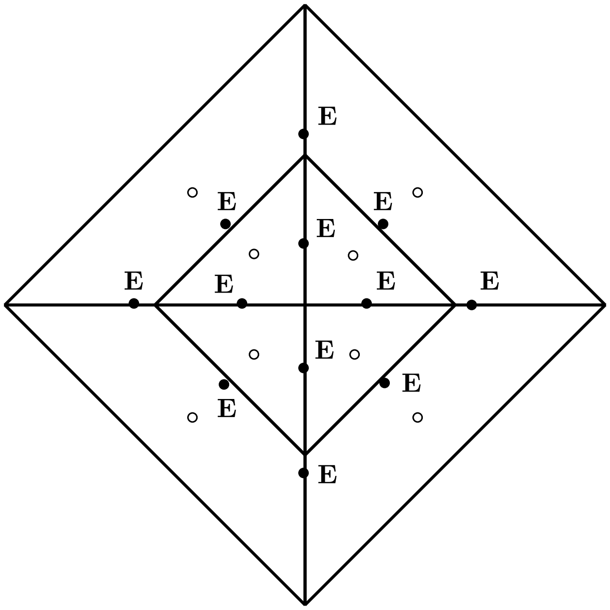
For an expanded example of this process consider the mesh illustrated below.



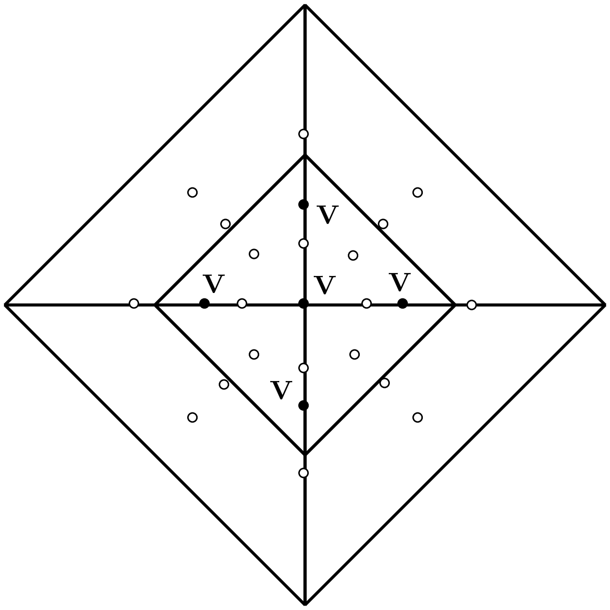
First, we construct the face points, which are calculated as the average of the points that make up each of the original faces. These points are shown in the figure below, labeled with an $\mathbf{F}$.



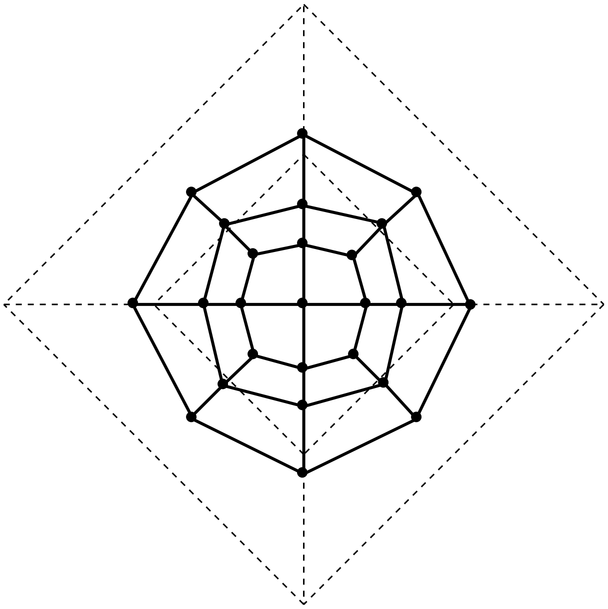
Now, we construct the edge points, which are calculated as the average of the four points – the two original points which define the edge, and the two new face points for the faces that are adjacent to the edge. These points are shown in the figure below and are labeled with an $\mathbf{E}$ (The face points calculated in the previous step are indicated as points with white centers).



We now construct the vertex points. These points are shown in the figure below, with the face points and edge points indicated with white centers.

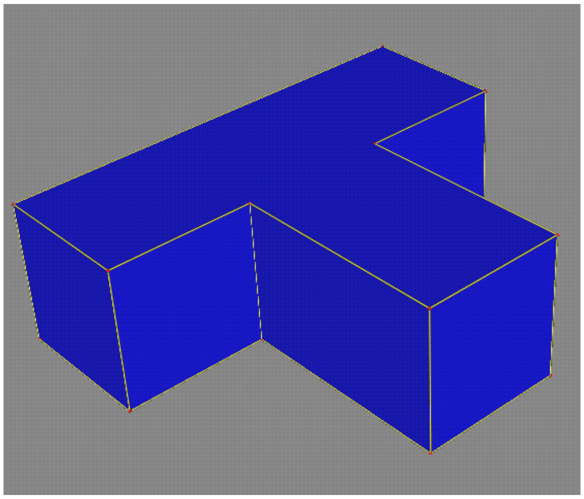


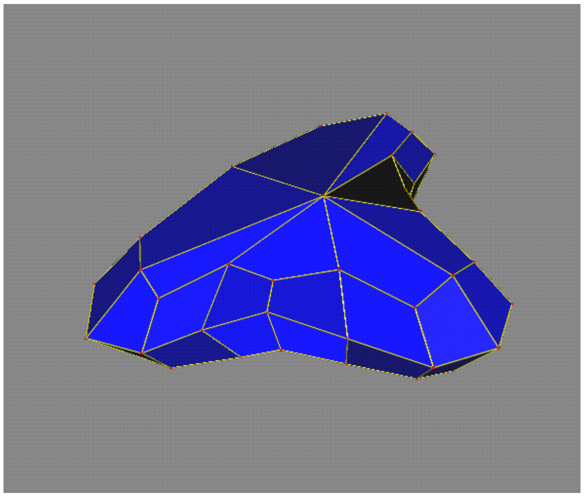
Finally, we connect the edges to the points we have generated: First connecting the face points to the edge points that correspond to edges on the face, and then connecting the vertex point to the edge points.



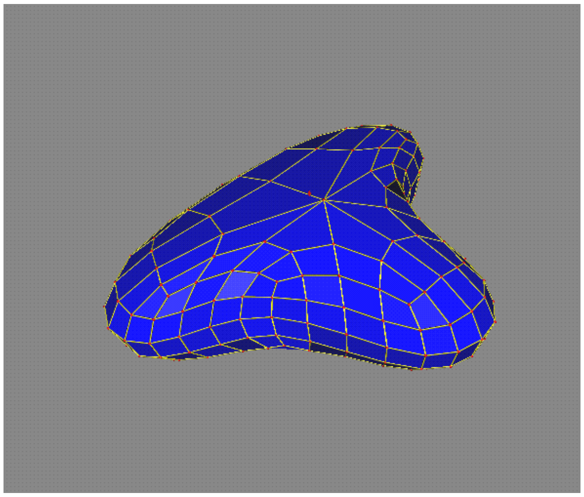
We note now that the each face of the refined mesh has four edges, and in fact, this is true in all cases no matter how many sides the original figure has.

Four steps in the Catmull-Clark refinement process are shown in the illustrations below. Note what happens to the corner control points under the refinement process.

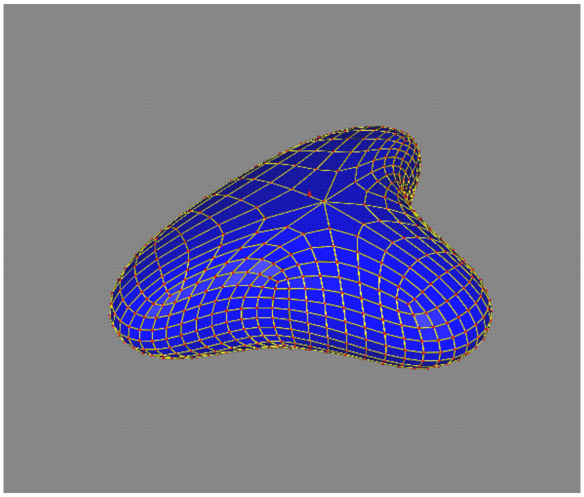




We note that the new set of control points has the property that all faces have four sides. However also note that the vertices corresponding to the original control points retain the *valence* (the number of edges that are adjacent to the vertex). One of the quadrilaterals is shaded incorrectly, since it is non-planar and the rendering algorithm used cannot process these correctly.



Notice still that the vertices corresponding to the original control points retain their valence. Notice the vertex of valence eight at the top of the solid.



Note that any portion of the surface where we have a 4×4 array of control points in a rectangular topology, represents a bicubic uniform B-spline surface patch. As subdivision proceeds, the only place where such a topology will not exist is at the vertices that represent the original control points, and whose valence is not four (commonly called extraordinary points). If all vertices in the original had valence four, we would have a bicubic uniform B-spline surface patch to start with.

Summary

Catmull and Clark have shown that the rules expressed for the cubic B-spline subdivision not only work for arbitrary rectangular meshes, but can also be extended to meshes of an arbitrary topology. This methodology produces a surface that is locally a bicubic uniform B-spline except at a finite number of points on the surface.

References