MA 323 Geometric Modelling
Course Notes: Day 35
Polyhedral Surfaces

David L. Finn

32 Polyhedral Surfaces and Subdivision Surfaces

In this chapter, we explore a different type of surface creation method. We start the chapter with a simple method based on a Hermite type interpolation for a triangular patch. This method is a extension of the methods in the previous chapters to a introduce a new structure for creating surfaces. In this structure, we start with a polyhedral surface, a collection of planar faces. We then define a new polyhedral surface from the old polyhedral surface, that is a subdivision algorithm.

More generally, in this chapter, we define a discrete approximation to a surface as a collection of points, edges, and faces. This is a direct generalization of a discrete curve as collection of line segments, that is a polyline or a for a closed curve a polygon. These surfaces are generalizations of the standard polyhedrons from Solid Euclidean geometry; tetrahedrons, cubes, hexahedrons, dodecahedrons, et cetera. For our purposes, we consider subdivision algorithms to produce refinements of a polyhedral surface, so that in the limit a smooth surface is created. A subdivision surface is a surface that arises by applying a subdivision algorithm (a recursive algorithm) to a polyhedral surface.

To describe our procedure more precisely, we recall that a polygon is an object in a plane that consists of a set of vertices (points) and edges where the edges are straight lines that join to points. More generally, a polygon is a special type of planar graph (from Discrete and Combinatorial Mathematics). For a polygon, the edges can be described as an ordered list of the edges, where edges are connected in order, and each pair of edges meets at a vertex. This implies since the list is closed the first vertex and the last vertex are the same that the number of vertices is equal to the number of edges in a polygon.

A polyhedral surface is a spatial object that consists of faces (typically a polygon in a plane), edges (line segments) and vertices. One of the principle goals of this chapter is to examine the structure of a polyhedral surface. This entails developing new data structures, as we will not be creating parametric surfaces. These data structures are especially nice, when one uses an object oriented programming language to describe the surfaces.

A second goal for this chapter is to introduce the notion of a subdivision surface. Recall, we talked about subdivision curves a little bit, when using de Casteljau's algorithm to create a Bezier curve. A subdivision surface is an extension of these methods. The advantage of subdivision surfaces over patches is that they are easy to implement on computers, as they are iterative to create a refinement of an object. Subdivision surfaces also are nice because you can define the object topologically, that is up to the number of holes in the surface. The subdivision algorithms preserve topological structure, which is nice. To create a topological surface with patches, requires significant work in order to glue the patches together in a nice manner. Subdivision surfaces remove that annoyance. Basically, one only has to create a nice "framework" for the surface. The disadvantage is that it is hard to shape the surface, and that designing with subdivision surfaces is time consuming if you have to specify the information directly. However, subdivision surfaces are efficient given that the initial framework can be sampled directly from the object that is to be modelled then subdivision surfaces are easy to work with. There are no equations to solve to arrange the data, and we can work directly from the data provided.

33 A Hermite-Type Interpolation for Triangular Patches

Consider the following problem:

> Find a smooth surface $\mathcal{S}$ that passes through a given collection of points $\{p_i\}$ in space and having prescribed normal vectors $\{n_i\}$ at the given points.

There exist powerful methods for arranging a given collection of points into triangles, so that it suffices to consider the problem for three points and three normal vectors.

We therefore consider three points with corresponding normal vectors (p_a, n_a), (p_b, n_b), and (p_c, n_c). Rather than producing an expression for every point on the surface, we define a new collection of triangles with points and corresponding normal vectors. The method we describe with produces four triangles from the one triangle. We do this by defining a point and normal vector on each edge curve and then forming new triangles as per the diagram below.

We define the point on the edge curve p_{ij} and the normal vector n_{ij} as follows:

- For $i \neq j$, define $v_i^j = p_j - p_i$ and then the tangent vector for the edge curve t_i^j at p_i as the orthogonal complement of the vector v_i^j to the normal vector n_i, that is

$$t_i^j = v_i^j - \frac{v_i^j \cdot n_i}{n_i \cdot n_i}\, n_i.$$

- For $i \neq j$, define $p_i^j = p_i + \frac{1}{3}\, t_i^j$.

- Define $p_{ij} = \frac{1}{2}\, p_i^j + \frac{1}{2}\, p_j^i$.
- Define $n_{ij} = \frac{1}{2}\, n_i + \frac{1}{2}\, n_j$.

Notice that in this definition $p_{ij} = p_{ji}$ and $n_{ij} = n_{ji}$, but $p_i^j \neq p_j^i$ since $v_i^j = -v_j^i$ and n_i is not necessarily equal to n_j, see diagram below. Furthermore, notice the points p_i^j and p_j^i are basically the Bezier control points for the Hermite curve through p_i and p_j with tangent vectors v_i^j and $-v_j^i$.

With these new edge points p_{ab}, p_{bc} and p_{ca}, we define the four new triangles by $\{p_a, p_{ab}, p_{ac}\}$, $\{p_b, p_{ba}, p_{bc}\}$, $\{p_c, p_{ca}, p_{cb}\}$ and $\{p_{ab}, p_{bc}, p_{ca}\}$. This process is a subdivision algorithm, as for each triangle we produce four new triangles and we refine the approximation of the surface. Each of the original three points that was on the surface remains on the surface. After the first iteration, we have six points on the surface. After the second iteration, we have eighteen points on the surface. The six from the first iteration, plus 3 more for each triangle. Therefore, the number of points on the surface after n iterations is $p_n = 3\,4^{n-1} + p_{n-1}$ as there are 4^{n-1} triangles in the $n-1$st iteration.

The table below shows the number of points and the number of faces in the approximation to the surface. From the data in this table, one easily sees that the number of points on the surface after n iterations equals $4^n + 2$.

n	number of points	number of faces
0	3	1
1	6	4
2	18	16
3	66	64
4	258	256
5	1026	1024

33.1 Exercises

1. Given a set of four points and four normal vectors as described in the diagram below. How can you apply the subdivision procedure in this section to obtain a surface that

passes through the given points and has the given normal vectors? Is there only one such surface?

2. Apply your solution to the previous problem to the data

$$\begin{aligned} p_a &= [0,0,0], \quad n_a = [0,0,1] \\ p_b &= [1,2,1], \quad n_b = [1/2, 1/2, 2/3] \\ p_c &= [1,3,0], \quad n_c = [-1/4, 2/3, 2/3] \\ p_d &= [-1,2,-1], \quad n_d = [1/3, -1/3, 9/10] \end{aligned}$$

3. Construct a subdivision algorithm, given a set of four points and four normal vectors given that the four points all lie on the same plane that is the natural generalization of the subdivision algorithm in this section.