Paper 1: Surface Meshing

Introduction

Surfaces of three-dimensional objects may be described in a number of ways. Some objects may be completely described by a mathematical function, such as that for a sphere with its centre at the origin:

$$r^2 = x^2 + y^2 + z^2$$

More complex surfaces may be described in a piece-wise fashion by a set of equations $f_i(x, y, z) = 0$, where each equation describes a part of the surface for some particular range of the variables x, y, and z. These equations may represent complex curved surfaces or may be as simple as planar elements, the complexity depends on the surface being described and the desired accuracy of representation.

To model the physical surface of a three-dimensional object with a set of equations requires that the object be measured. This measurement process normally consists of determining the locations of a set of points on the surface of the object with respect to a reference coordinate system and recording the way in which these points are mutually associated (which points belong to which piece of the surface). These two data sets (the point locations and the connectivity information) can then be used to construct a set of equations, $f_i(x, y, z) = 0$, which describe the surface.

This chapter describes a method by which a surface approximating an object can be developed from a set of points digitized from that object without knowledge of the way in which the points are organized on the surface. The method reconstructs the organization among points to produce a surface described by a series of triangles.

Definitions and Comments

A point is a location on the surface of the object and is defined by a Cartesian coordinate triplet (x, y, z) relative to an arbitrarily selected origin. Points will be referred to by capital letters like P. Subscripts will be used when it is necessary to indicate a particular point among a group of related points as in: P_i. The coordinates of a point P_i may be designated by (x_i, y_i, z_i) or when needed separately as the entities: x_i, y_i, and z_i.

A point list is a set of points that is used to describe an object. The points may appear in any particular order within this list. All points in the list must be unique. A point list is denoted as a capital letter with a prime mark, such as: L'. A point list with n points in it is an $n \times 3$ matrix of point coordinates and may be displayed like:

$$L' = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_{n-1} \\ P_n \end{pmatrix}, \qquad L' = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ \vdots & \vdots & \vdots \\ x_{n-1} & y_{n-1} & z_{n-1} \\ x_n & y_n & z_n \end{pmatrix}. \tag{1,2}$$

The nearest point P_k from a point list L' to any given point A is that point whose Euclidian distance $|\overline{P_k A}|$ is the minimum of all other points in the list L'.

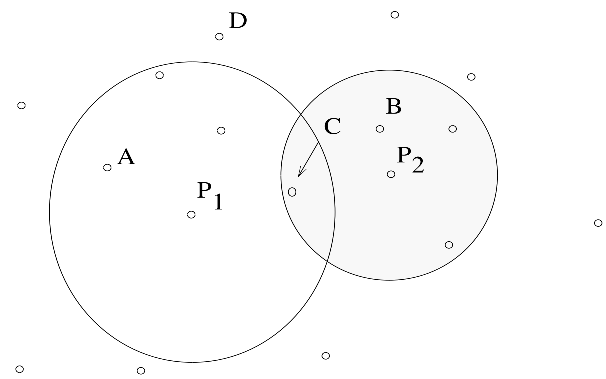

Figure 1: Local points

A local point is any point that is in the immediate neighborhood of a point A. The extent of the immediate neighborhood surrounding A is defined by considering it to be a list L' of m points that are the m nearest points to A. These local points play an important role in increasing the speed of surface construction. The actual number, m, of points that are considered local to a given point A, and its effect on the surface construction will be considered later. Figure 1 (where $m = 4$) illustrates several aspects of local points. First the physical extent of the local neighborhood depends on the number of points included in it, and it may not be the same for two reference points (as is the case for P_1 and P_2). Second, for any two given reference points (P_1 and P_2) there may be points (for example C) local to both. Finally, local points will not necessarily be uniformly distributed in the immediate neighborhood as is demonstrated by the distribution of local points about P_1.

An edge is a line connecting two points P_i and P_j. An edge is usually denoted by: $\overline{P_i P_j}$, this notation also implies a vector $\vec{v}$ directed from P_i to P_j. Associated with each edge are two other nearby peripheral points (T_k and T_m in Figure 2) which define the two triangles sharing the edge. Edges also have some peripheral data that is used to keep track of the exterior of the object being reconstructed.

All edges have two and only two end points. When a group of $n \geq 3$ points are found to be co-linear these n points will be treated as defining $n - 1$ edges rather than just one edge. Once the surface is completed all edges will have two and only two triangles attached to them. These two triangles are defined by the two points P_i and P_j on the edge which they share and the two points T_k and T_m that are nearby (defined below) to both of the edge's end points. It is for this reason that the terms: "nearby points" and "triangles" may be used somewhat interchangeably.

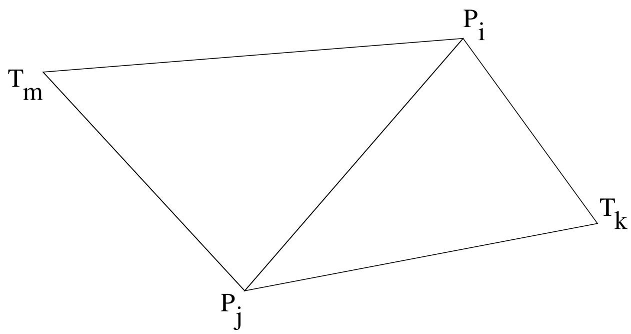

Figure 2: Two triangles sharing an edge

A nearby point, in contrast to a local point, is a point P_i that shares an edge with a given point A. As a result of the triangular surface patches used to create the surface, any given point A must have at least 3 nearby points; however, there is no upper limit to the number of nearby points. This is illustrated in figure 3 where point P_1 has 6 nearby points and point P_2 has 3.

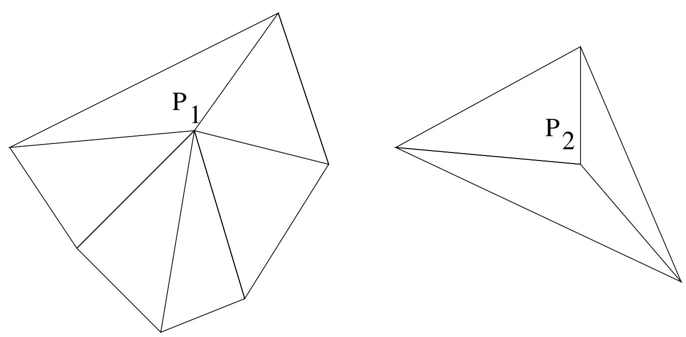

Figure 3: Nearby points

A triangle is the basic unit used to construct the finished surface. A triangle as used here follows the normal mathematical definition of being a planar figure with three straight sides. Each side of a triangle is an edge. Each of these edges will be shared with one and only one other triangle. If more than two triangles were attached to one edge, a complex surface that is not piecewise continuous (such as that shown in Figure 4) would be formed.

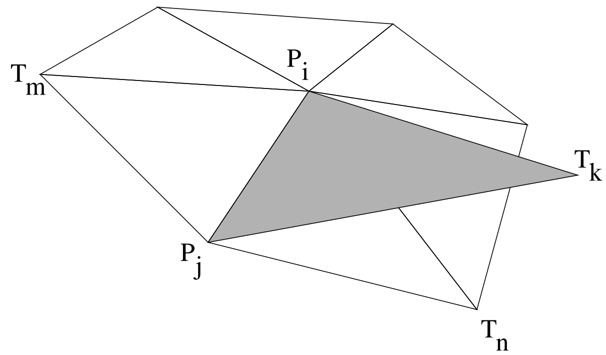

Figure 4: A Non-continuous surface formed by an edge with 3 triangles

Each triangle will contain three points from the object's point list. These points will be shared with other triangles but there is no fixed number of triangles sharing each point, only a minimum of 3 triangles must share each point. Triangles do not intersect apart from the sharing of complete edges. In the case where a pair of edges would intersect (for example joining the four corners of a square by the diagonals) only one of the edges will be used so that edges never intersect.

The exterior of a triangle is the set of points in the plane of the triangle that are not contained within the three line segments that form the sides of the triangle.

The interior of a triangle is the set of points in the plane of the triangle that are contained within the three line segments that form the sides of the triangle.

The vectors $\vec{i}, \vec{j}, \vec{k}$ are the unit in the directions of the Cartesian coordinate system axes. The vector $\vec{i}$ is the unit vector in the direction of the x-axis, $\vec{j}$ is the unit vector in the direction of the y-axis, and $\vec{k}$ is the unit vector in the z-axis.

The outside of a triangle (in contrast to the exterior) is a quantity that defines on which side of the triangle's surface the space surrounding the object lies. Establishing this quantity for each triangle in the object's surface description is essential for a number of computations based on volume integrals that appear later in this work. The outside of the triangle can be described explicitly by a vector centered at the triangle's center of gravity that points in the direction of free space. It can also be described by a right hand rule based notation. If a vector pointing in the direction of free space exists then by grasping that vector in the right hand with the thumb pointing towards free space the fingers will curl in a counterclockwise direction in the plane of the triangle. This leads to the convention that the vertices of the triangle will be recorded in a counterclockwise order, so that by curling the fingers in the order that the triangle's vertices appear the thumb will point towards free space. This is illustrated in Figure 5.

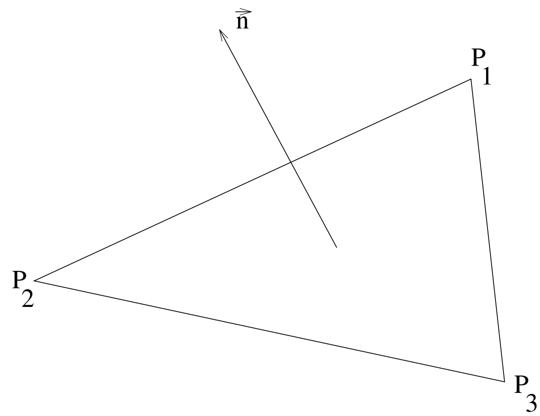

Figure 5: Outward normal direction vector

This also allows the outward pointing normal vector $\vec{n}$ for any triangle $\triangle P_1P_2P_3$ to be written as:

$$\vec{a} = \overline{P_2P_1}, \quad \vec{b} = \overline{P_3P_1}, \quad \vec{n} = \vec{a} \times \vec{b} \tag{3,4,5}$$

wherein $\times$ is the vector product (cross product) operator and $\vec{a} \times \vec{b}$ may also be written:

$$\vec{a} \times \vec{b} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{vmatrix}. \tag{6}$$

The inside of a triangle (in contrast to the interior) is a quantity that defines on which side of the triangle's surface the space within the object lies. If the outside direction vector for a particular triangle is $\vec{n}$, then the inside direction vector is just $-\vec{n}$.

A cost function is a function to rank points when creating new triangles by adding points to existing edges. A cost function is defined in terms of the coordinates of the point to be tested (P_i) and the points that define the edge (P_1 and P_2 forming edge $\overline{P_1P_2}$) that a new triangle is being built from. Cost functions are defined such that a smaller value indicates a more desirable point. Cost functions are denoted by a "C" with a subscript that indicates the particular cost function. An example is

$$C_0(P_1, P_2, P_i) = |\overline{P_1P_i}| + |\overline{P_2P_i}| \tag{7}$$

which is the minimum added perimeter cost function. Here the function's arguments are given as points, but they can also be written explicitly as the points coordinates like:

$$\begin{aligned} C_0(x_1, y_1, z_1, x_2, y_2, z_2, x_i, y_i, z_i) = & \\ & \sqrt{(x_1 - x_i)^2 + (y_1 - y_i)^2 + (z_1 - z_i)^2} \\ & + \sqrt{(x_2 - x_i)^2 + (y_2 - y_i)^2 + (z_2 - z_i)^2} \end{aligned} \tag{8}$$

The fold angle is defined as the angle between the two outside normals to the two triangles that share an edge. This is illustrated in Figure 6 where the fold angle between the $\triangle P_iP_jT_m$ and the

$\triangle P_i P_j T_n$ at the shared edge $\overline{P_i P_j}$ is desired. The outside normal vectors to the two triangles under consideration are determined from equation 6. The fold angle is then determined from the two normal vectors $\vec{N}_1$ and $\vec{N}_2$ from the dot product (scalar product) of $\vec{N}_1$ and $\vec{N}_2$:

$$\cos\theta = \frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{|\vec{N}_1||\vec{N}_2|} \tag{9}$$

In general, once the first triangle in the surface has been defined each point that is added to the surface will introduce at least one new triangle, the fold angles between this triangle and any triangles that it shares edges with must be within the allowable fold angle limits.

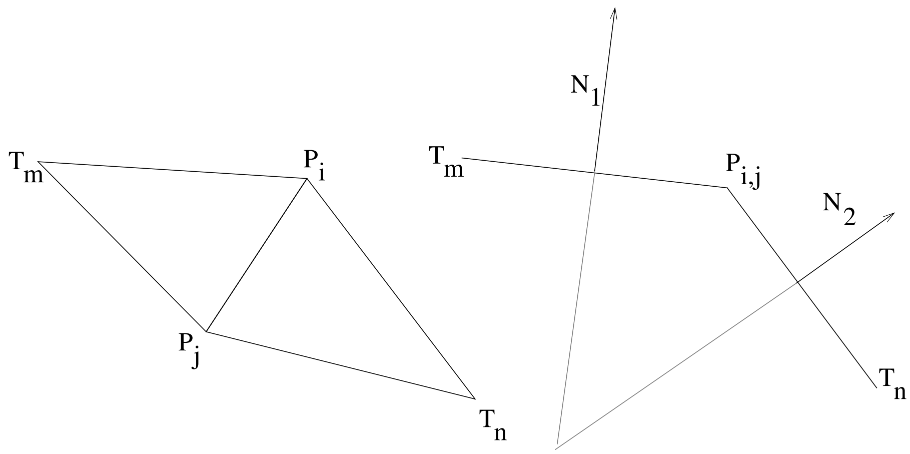

Figure 6: The fold angle

By limiting the allowable range of values with a limiting fold angle it is possible to control how fast the surface will bend. In this implementation this fold angle parameter is a user-definable constant that must be manually tuned. It is possible that an automatic tuning procedure or an alternative to the fold angle could be developed.

Related to the fold angle is the issue of how dense the digitized point data should be packed on the surface. This data should be packed closely in areas where the object being digitized curves sharply, while it can be relatively sparse in areas where the curvature is not great. There is an additional complicating factor: the thickness of the object. The working rule here is that the spacing between points should be less than the thickness of the object, otherwise there is the possibility that the algorithm will pick points on the other side of the object as being nearby causing a pinching off effect. This rule assumes a maximum fold angle of 45 degrees is in effect, for smaller limiting fold angles the spacing between points may be increased for a given object thickness.

A grid of points is a list of points wherein the connectivity between points is arranged in some regular repeating pattern. Grids typically arise when a surface has been scanned by a machine which follows some predefined path about the object.

A mesh of points is a list of points with the connectivity data (point to point) defined. This forms a mesh or net surrounding the object but does not form a surface since the additional information