Feature-based Decomposition of Trimmed Surface

WU Yiu-Bun

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Automation and Computer-Aided Engineering

September 2004

Abstract

Trimmed surface is one of the most commonly used elements in CAD/CAM systems. In general, a trimmed surface is represented by a parametric surface and a group of trimming curves specified in the parametric space of the surface. Because of the complexity in manipulating trimmed surface, some CAD processes and algorithms cannot be applied to trimmed surface directly. Thus, it is desirable to represent a trimmed surface by a group of simple regular surfaces, whose union corresponds to the trimmed surface. A decomposition algorithm which performs the task will be presented in this thesis.

The algorithm consists of several different techniques. In the parametric space of the surface, Voronoï diagram is first developed so that bisectors are derived to define an isolated region for every trimming curve. Switching to the three-dimensional space, feature points are determined on every trimming curve. Correspondence will be established between the feature points and the bisector vertices according to the similarity between the shapes of the trimming curves and their bisectors. Regions of four- or three-sided parametric patches are then identified. Finally, regular surfaces are constructed by interpolating the sampled surface points on each of the identified regions. The result is a group of regular surfaces resembling the original trimmed surface.

In this thesis, the details of each step are described and explained with worked examples. The strength and weakness of the algorithm are also analyzed by comparison with a similar tessellation algorithm.

摘要

截裁曲面（Trimmed Surface）在計算機輔助設計及製造系統中被廣泛使用。一般而言，一個截裁曲面由一個參數化表面（Parametric Surface）及一組在參數化空間定義的截裁曲線（Trimming Curves）所代表。由於處理截裁曲面的複雜程度，一些有關計算機輔助設計的程序和算法不能直接應用在截裁曲面之上。因此，我們需要利用一組簡單而正規的曲面來代表截裁曲面，而該組曲面的併集跟原本的截裁曲面是相對的。本論文會提出一個爲達到以上目的而將截裁曲面解體的算法。

此算法由幾種技巧組成。在曲面的參數化空間裡，首先會建立一個 Voronoï Diagram，其平分線（Bisector）會爲每一條截裁曲線界定一個隔離的區域。在三維空間裡，該算法會找出存在於每一條截裁曲線上的特徵點（Feature Points）。根據截裁曲線及其平分線在形狀上相似的程度，平分線上的點和特徵點之間的對應（Correspondence）從而確立。然後，由四條或三條邊劃分成的區域會被認定爲參數化補片（Parametric Patches）。最後，在這些區域相對的曲面區域上抽取的樣本點進行補插（Interpolation），便可以建立出正規的曲面。結果是產生出一組用來比擬原本截裁曲面的正規曲面。

本論文會爲每一個步驟的細節加上不同的實例進行闡釋以及描述。在論文中提出的算法會跟相似的密鋪（Tessellation）算法作出比較，籍此分析其利弊。

Contents

Chapter 1. Introduction

Trimmed surface plays an important role in CAD/CAM technology. It is represented by a surface with its control data in the three-dimensional space, as well as the trimming curves in the parametric space. However, there is no standard representation for trimmed surface among different types of CAD systems. The differences between the systems results in difficulties for exchanging trimmed surface data. Moreover, there are geometric algorithms that cannot be directly applied to trimmed surface. For example, trimmed surfaces are usually the result of intersecting different surfaces. Therefore, deformation of a trimmed surface may require re-evaluating the trimming curves which is a time consuming process.

For these reasons, one solution is to decompose trimmed surface into other types of basic standard elements which can be easily recognized by most CAD systems. Decomposing the trimmed surface into patches is one of the feasible solutions. The union of these patches resembles the original trimmed surface. The algorithm presented in this thesis is designed to decompose a trimmed surface into surface patches. The distribution and the shapes of the trimming curves are considered in the decomposition process. The output of the algorithm is a group of connected non-intersecting surface patches. The patches contain no trimming curves and can be represented as simple surfaces.

Chapter 2. Previous Works

There are numerous researches about decomposing or tessellating trimmed surface. Typically, they can be classified into two main approaches. They are surface patch approach and triangular facet approach.

Surface patch approach develops a group of surface patches to represent a trimmed surface. The patches are usually expressed in the form similar to that of the trimmed surface, such as Bézier, B-Spline or NURBS surfaces, but without any trimming curves on them. The main issue of this method is to design an algorithm that can reasonably divide the trimming curves in segments, which constitute parts of the boundaries of the patches. In this way, there is no trimming curve on the decomposed surface, and the union of the patches resembles the shape of the original trimmed surface. The algorithm presented in this thesis belongs to this category.

By triangular facet approach, the surface is divided into a group of triangular facets. Usually, a physical tolerance is specified in the division process so that the triangular facets approximate the surface with deviation not exceeding the tolerance. The division is usually carried out in the 3D space or a specially defined domain so as to keep the approximation accurate.

2.1. Surface Patch Approach

Several research work on converting trimmed surface into surface patches will be discussed in this section. In the algorithms proposed by these work, the critical step is to convert the parametric trimming curves into curve segments. These segments act as parts of the boundaries of the surface patches. The other parts of the boundaries maybe composed of line segments, for example, scan-lines which are segmented at their intersections with the trimming curves. Every parametric patch usually consists

of four or three sides in its boundary. Interpolating the sampled surface points will form the patches in the physical space. Thus the surface patches contain no trimming curves in their valid region. The original trimmed surface is resembled by the patches union.

Hamann and Tsai [1] proposed an algorithm which tessellates a trimmed NURBS surface by the union of a set of ruled surfaces. Voronoï diagram is first developed by triangulating the parametric space of the trimmed surface. During the triangulation, right-angle triangles are regularly tiled to cover the whole parametric space. Then on every triangle edge, a point which is equidistant from the two nearest parametric trimming curves is determined. Sorting and joining these points together forms end-to-end line segments called bisectors. The bisectors bisect the valid region between any two adjacent trimming curves, and thus define an independent region for each trimming curve in the parametric space. At the intersections with the bisectors, the boundary curve is broken down and connected with the bisectors so that every trimming curve is surrounded by the corresponding closed bisector and the boundary segments.

In the above process, the valid region between each trimming curve and its closed bisector and boundary segment is defined as parametric tile. There is only one trimming curve in a tile. Extreme points of the tiles, such as trimming curve points with zero derivatives or local maxima and minima on the bisectors, are determined. Horizontal scan-lines are then emitted from these extreme points until they reach the bisector or the boundary segment of the tile. The tile is "cut" along these scan-lines and the region being divided contains no trimming curves. These regions are parametric patches and the boundary of a parametric patch is composed by two scan-lines, a bisector segment and a boundary segment.

After all the partitioning processes are finished, ruled parametric surfaces are generated by applying uniform linear interpolation to sets of points sampled within the parametric patches. Mapping the patches from the parameter space to the 3D space of the trimmed surface, ruled surface patches are developed. The union of the patches then resembles the valid part of the trimmed surface.

The algorithm by Vries-Baayens and Seebregts [2] converts trimmed non-rational Bézier surface into composite or basic surfaces, such as coons surfaces, which are interpolated linearly. A technique for decomposing non-rational Bézier surface based on the combined 'triangulation-quadrangulation' of a trimmed surface is applied in the parametric space. The resulting surfaces are enclosed by the approximation of the boundary while the surfaces lie exactly in the original basic surface in the 3D space. There are cases, such as curve with inflection points and curve degeneration, that the interpolation may fail to keep the exactness of the resulting surfaces. Solutions to these problems are classified and discussed in the reference.

2.2. Triangular Facet Approach

In this section, a number of triangular facet approaches are listed and their approaches are briefly described.

In the algorithm by Piegl and Richard [3], a physical tolerance is specified and the valid region of a trimmed NURBS surface is triangulated in the parametric space. The parametric triangles are projected onto the 3D space to evaluate the deviation between the approximating triangles and the real model. The triangulation is recursively by until the deviation is less than a preset tolerance. A piecewise planar approximation of the trimmed surface is obtained.

Cho *et al.* [4] converted trimmed rational B-Spline surface patches into triangular mesh generated by an algorithm based on the unstructured Delaunay mesh approach. The algorithm constructs a two-dimensional domain where the triangulation of the patches carries out. The domain ensures the shapes of the generated triangles to have sufficient preservation while mapping into the 3D space. The topological consistency between the original surface and the approximating mesh is thus maintained. The linear approximation of the boundary curves and the initial triangulation are performed initially. Further triangulation follows, within a given approximation tolerance.

Abi-Ezzi *et al.* [5] made use of the graphical data compilation concept to dynamically tessellate a trimmed NURBS surface. The algorithm is implemented for fast and dynamic use. The process is divided into three stages. The data compilation phase accepts the surface geometry and outputs another form which is independent of the view. In the algorithm intensive phase, face culling, transformation and the tessellation step size is determined. Surface evaluations and triangular facet generation are performed in the compute-intensive phase. The research also studied how to avoid gaps between adjacent patches due to their different step sizes.

The algorithm proposed by Liu *et al.* [6] splits the irregular trimmed shapes into regular convex regions in the parametric space. The splitting process is performed in the parametric space by determining the intersections between the horizontal and vertical scan-lines and the boundaries of the irregular trimmed shapes. The extreme points of the shapes are determined. According to different cases of the intersections, the irregular shapes are divided into segments at the intersections. The splitting process continues until the shapes of all regions are regular with no extreme points. These regions are then tessellated into triangular facets under a certain accuracy defined by the users.

A geometry-based algorithm is used by Piegl and Tiller [7]. A trimmed NURBS surface is tessellated into triangular facets based on its geometric characteristics. The surface does not have to be C^1 continuous. The triangulation is carried out in the parametric space and the subdivision is performed in the 3D space. The subdivision is independent of the surface's parameterization.

Cho *et al.* [8] constructs an auxiliary planar domain for the triangulation of the parametric trimmed surface. The mapping between the triangulation domain and the surface domain is performed by minimizing the mapping error function. The algorithm also determines if any approximating triangle intersects each other. In such case, the triangulation is repeated until no intersection occurs.

Triangulation gives a mesh of triangular facets ready for rendering and visualization. However, the triangular facets only gives an approximation to the trimmed surface, further editing of the surface will require manipulating the triangle vertices which may not be desirable.

Chapter 3. The Decomposition Algorithm

Although Hamann and Tsai's method effectively decompose a trimmed surface into regular surface patches, the patches created are horizontal strips in the parametric space of the surface. This may lead to an excessive number of patches. It is because the partitioning of the surface does not take the shape and features of the surface and the trimming curves into consideration. The method introduced in this thesis extended Hamann and Tsai's approach to decompose a trimmed surface by considering the shapes and features of the surface and its trimming curves. There are four stages in this algorithm, namely, Voronoï diagram development, feature point determination, vertices correspondence establishment and surface fitting. The result is a group of surface patches approximating the original trimmed surface, without irregular shapes on their boundaries.

3.1. Input to the Algorithm

The algorithm works on B-Spline surfaces but it can also be applied to other types of tensor product surfaces such as Bézier surfaces and NURBS surfaces. For the surfaces in 3D space, including the trimmed surface and the developed patches, the expression of a data point is given by

$$\begin{aligned}\mathbf{S}(u,v) &= (x(u,v), y(u,v), z(u,v)) \\ &= \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j} \,, \\ & \qquad u \in [u_0, u_1], \;\; v \in [v_0, v_1] \end{aligned} \tag{3.1}$$

where $\mathbf{P}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j})$ is a set of $(n+1)\times(m+1)$ of three-dimensional control points, $N_{i,p}(u)$ is the *i*-th B-Spline basis function of order *p* in the *u*-direction, u_0 and u_1 are respectively, the minimum and maximum knots of the knot vector in the *u*-direction. The basis function, minimum and maximum knots in the *v*-direction are defined in a similar way.

In the parametric space, a trimmed shape on the surface is a closed loop of curve segments. Such trimmed shape is called "trim" and the curves composing the trim is called trimming curves. In this thesis, trimming curves are B-Spline curves but other types of parametric curves can also be used. Denote the *i*-th trim as T_i, the *j*-th trimming curves on the *i*-th trim can be expressed as

$$\begin{aligned}\mathbf{C}_{i,j}(t) &= (u(t), v(t)) \\ &= \sum_{k=0}^{n} N_{k,p}(t)\mathbf{P}_k, \\ & \qquad t \in [t_0, t_1] \end{aligned} \tag{3.2}$$

where $\mathbf{P}_k = (u_k, v_k)$ is the $(n+1)$ two-dimensional control points of the curve, $N_{k,p}(t)$ is the *k*-th B-Spline basis function of order *p*, t_0 and t_1 are respectively the minimum and maximum knots of the curve's knot vector. In most case, each curve on a trim is C^1 continuous and every trim must be C^0 continuous. Every two adjacent curves share only one common end point and thus every trim is closed. By the definition above, the boundary trim is denoted as T_0. *Figure 3.1* shows an example of a trimmed surface in parametric space and its physical model is shown in *Figure 3.2*. It can be noticed that the curves on the boundary trim is arranged in anticlockwise order while the other trims are arranged in clockwise order. Every trim is closed and the dots on the curves in *Figure 3.1* represent the end points of the curve

segment. The valid part of the trimmed surface is defined by the region between the boundary trim and the other trims. The number of trims and the number of trimming curves are arbitrary. The shapes of the trims are also arbitrary. Therefore, the boundary trim may not coincide with the boundary of the parametric space. In the parametric space of a surface, the coordinate of the left lower corner in the parametric space is (u_0, v_0) and that of the right upper corner is (u_1, v_1). The space is a rectangle rather than a square if the following condition holds

$$u_1 - u_0 \neq v_1 - v_0 \tag{3.3}$$

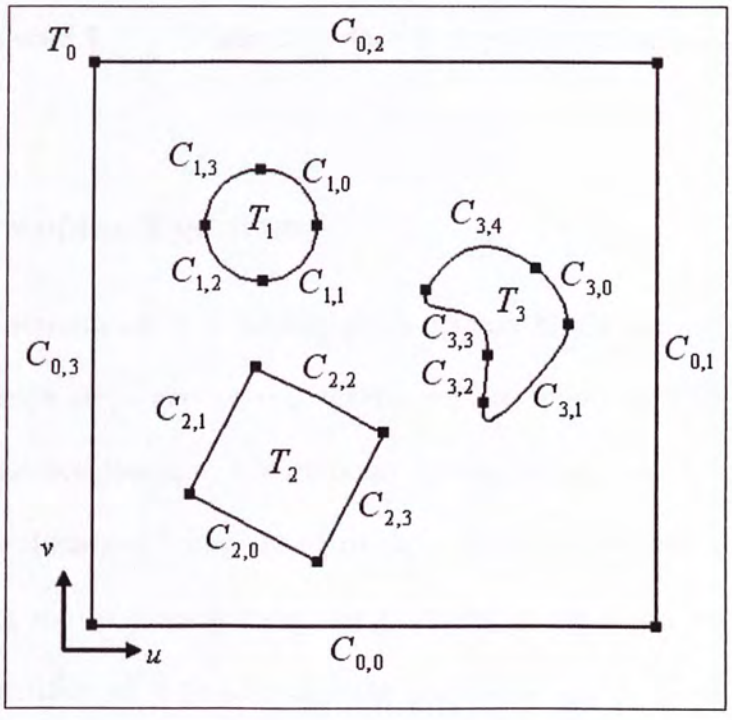

Figure 3.1 Trimmed surface in parametric space

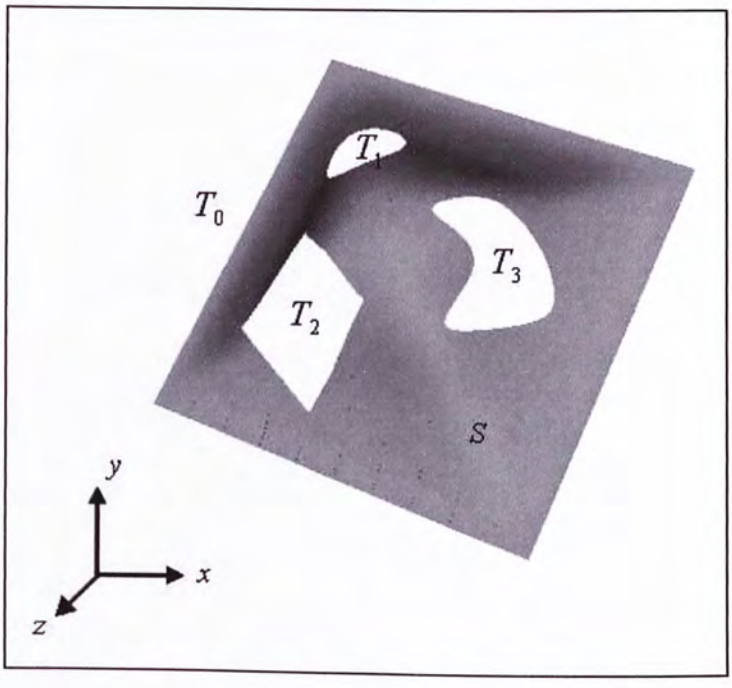

Figure 3.2 Trimmed surface in three-dimensional space

3.2. Overview of the Algorithm

The algorithm introduced is a surface patch approach. To decompose a trimmed surface, four main steps and several techniques are involved. The four steps are Voronoï diagram development, feature point determination, vertices correspondence establishment and surface fitting. The first three steps partition the valid part of the trimmed surface into regions without any trimming curve in the parametric space. Although the partitioning is processed in the parametric space, the shape features of the surface are also considered. These surfaces approximating the regions are called patches, and are the final result of the algorithm.

3.2.1. Voronoï Diagram Development

In the first stage, individual region must be defined for every trim, including the boundary trim. Therefore, Voronoï diagram is developed in the parametric space. The diagram consists of a series of straight line segments called bisectors. These line segments are connected one by one to define a closed region for every trim. The bisectors are midway between any two neighbouring trims. With the aid of the bisectors, every trim is isolated from its neighbouring trims. The valid region defined between a trim and its bisector loop is called a parametric tile. Further computations and analysis will be done on each parametric tile independently to generate the patches. The Voronoï diagram for the model in *Figure 3.1* is shown in *Figure 3.3*. The trims are in blue and the bisectors are in red. The dots on the bisectors in the figure indicate segment end points of the bisectors. They are called bisector vertices. Taking the boundary trim into account, there are four tiles developed for the four trims in the example.

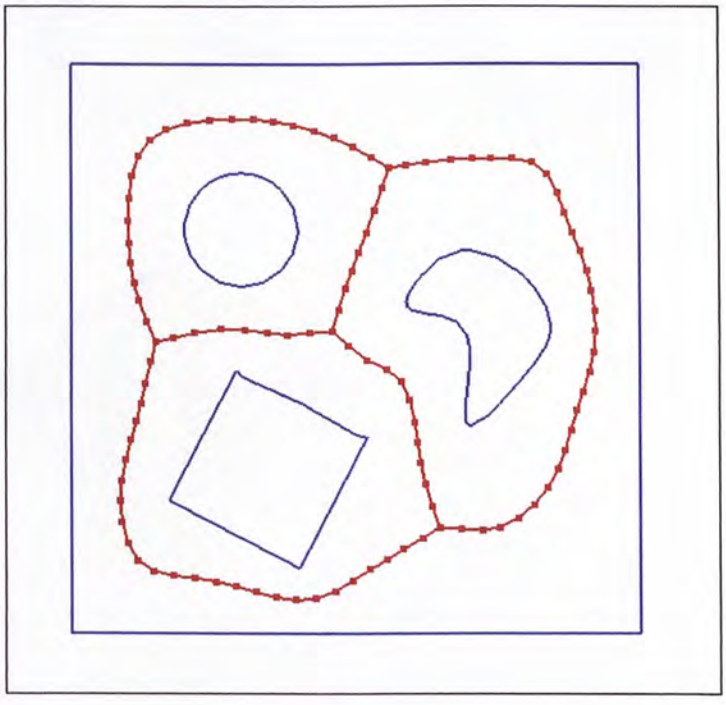

Figure 3.3 Voronoï diagram

3.2.2. Feature Point Determination

In the second step, the points where the trims divided are located. The divided trim segments will be the boundaries of the different patches. In order to minimize the irregularity of the patches, points with small radius of curvature are determined. These points are called feature points. A feature point can also be found at the junction between two consecutive curves with a sharp turning angle at the joint. The feature points are determined in the three-dimensional space instead of the parametric space. This is because a parametric feature point may not be a physical feature point. The parametric coordinates of the feature points are computed and will be used in the subsequent stages. In *Figure 3.4*, the green dots represent the feature points in the three-dimensional space and the parametric space. The model is displayed in different views.

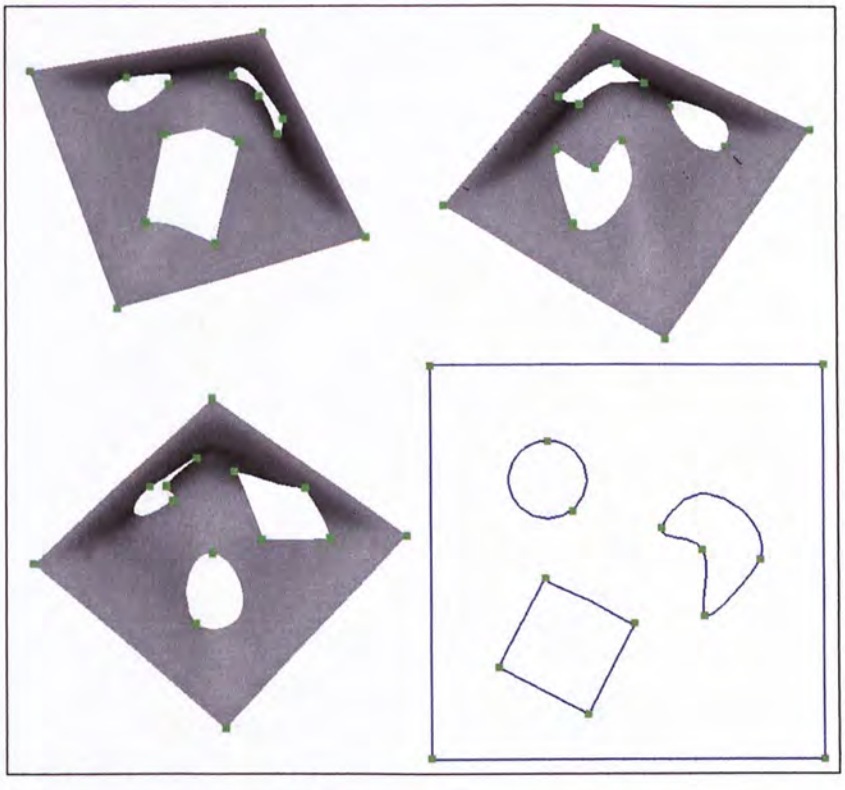

Figure 3.4 Feature points

3.2.3. Vertices Correspondence Establishment

Since the shapes of a trim and its bisector loop should have a certain degree of similarity, correspondence is established by applying the concept of shape blending. The process of establishing correspondence is to link feature points on the trims to the similar vertices on their bisectors. A ranking process is used to find the bisector vertex with highest degree of similarity. To measure the similarity, the shapes of the trims and their bisectors are normalized to a unit square space. The ranking method considers the distance between the normalized feature points and the normalized bisector vertices. In the ranking process, the interior angles at the bisector vertices in the 3D space are also considered. Thus, every feature points should finally be corresponded to the nearest and sharpest bisector vertices. The virtual line connecting a feature point to its similar bisector vertex is called a correspondence link, or simply link. The green lines joining the feature points and the bisector vertices in *Figure 3.5* represent the links for the model in *Figure 3.1*.

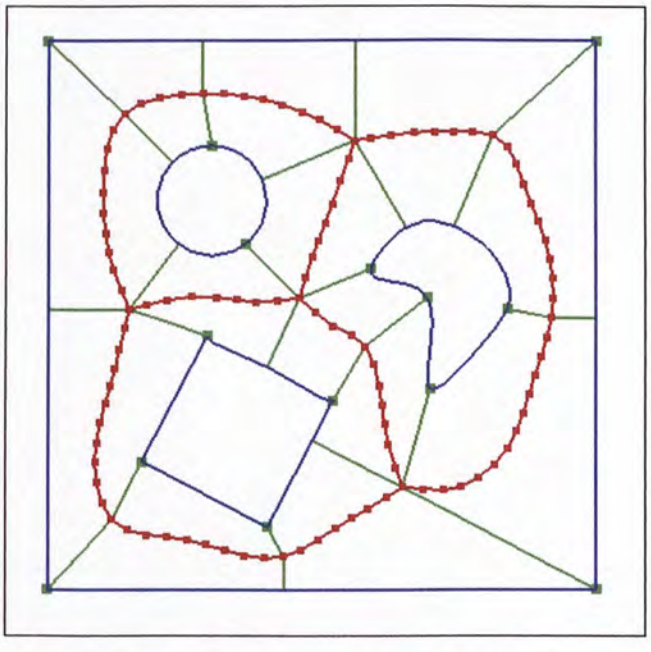

Figure 3.5 Correspondence Links

3.2.4. Surface Fitting

After all the partitioning computations are finished, the trims are divided into curve segments at the feature points. Similarly, the bisectors are divided into segments at the bisector vertices which are linked to the corresponding similar feature points. The tiles are then decomposed into regions with four sides as their boundaries. The four sides consist of a curve segment, a bisector segment and two links. These regions are known as patches in the parametric space, or called parametric patches. Interpolations are carried out on every two opposite sides of the patches. The control points of the patches in the 3D space, or called surface patches, are obtained. Each surface patch contains no trimming curve segments. Their union is an approximation of the original trimmed surface. *Figure 3.6* illustrates the interpolations within the parametric patches.

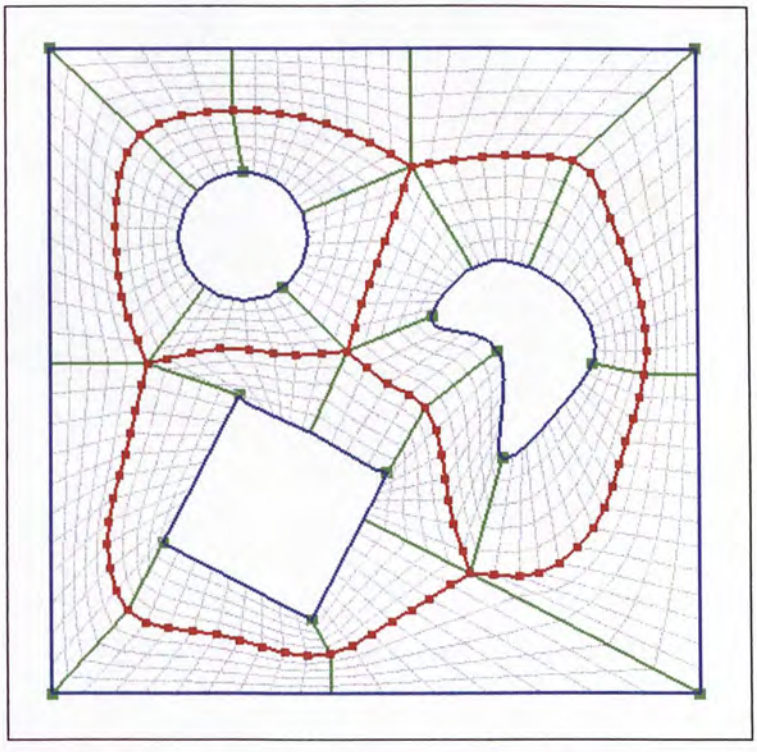

Figure 3.6 Parametric patches

3.3. Output of the Algorithm

The final output of the algorithm is the surface patches in the 3D space. The patches are formed by interpolating between the trimming curve segments, bisector segments and the correspondence links. The representation of the patches is the same as the input, i.e., the trimmed surface. They are represented as B-Spline surfaces in the 3D space. The difference is that the generated patches do not contain any trimming curves within their valid regions. Similar to the input, the surface patches can be represented by other types of tensor product surfaces such as Bézier surfaces and NURBS surfaces. *Figure 3.7* shows the final output of the model in *Figure 3.2*. The patches are coloured differently to indicate their existence.

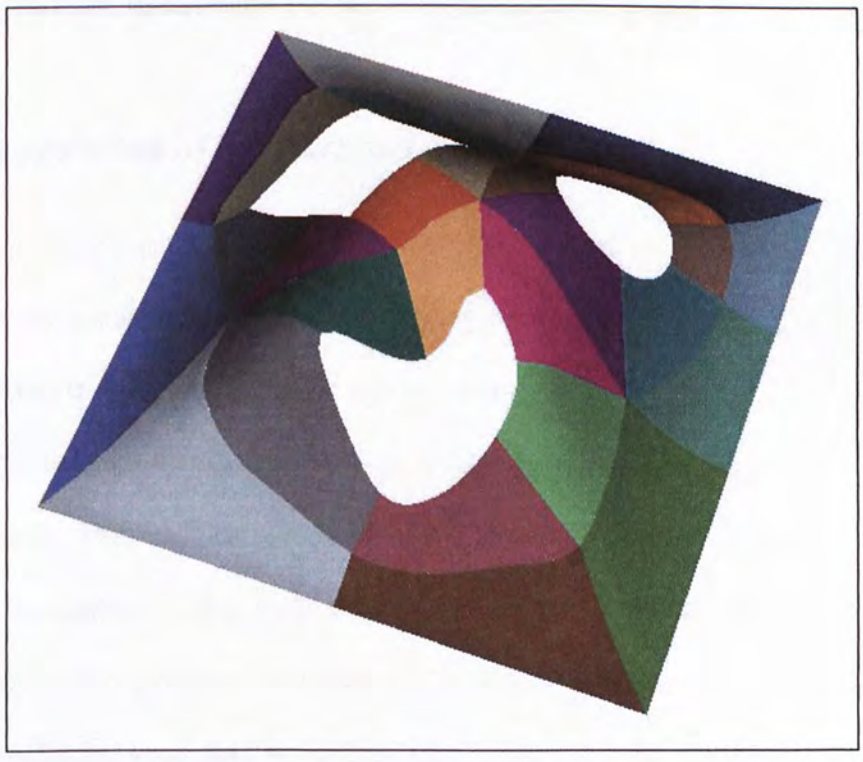

Figure 3.7 Final output of the algorithm

Chapter 4. Voronoï Diagram Development

The Voronoï diagram is developed for the trims so that every trim will have its individual region defined by its closed bisector. A Bisector is actually a series of contiguous straight line segments. They bisect the valid region of a parametric trimmed surface between two neighbouring trims. The technique for developing the Voronoï diagram is a modification of Hamann and Tsai's algorithm. In Hamann and Tsai's version, the bisectors are used to bisect the valid region for all the trims but not the boundary trim. In this thesis, the developed bisectors bisect the valid region for the boundary trim as well. Taking into consideration the boundary trim, every trim will have their own bisectors and thus their own valid region. The reason of including the boundary trim will be explained in the latter part of this chapter.

4.1. Triangulation of the Parametric Space

To develop the Voronoï diagram, a triangulation is first constructed for the trimmed surface in the parametric space. Here, triangulation means a number of right-angle triangles tiled in the rectangular or square parametric space regularly. The triangular tiles should completely cover the whole space. As an example shown in *Figure 4.1*, the thin grey lines are the edges of the triangular tiles. If a bisector exists and intersects an edge of a triangular tile, the intersection will be determined. Since the intersection is also a point on the bisector, it can be determined by using the fact that it is equidistant between the two nearest trims. Such point on the bisector is a bisector vertex. The bisectors can then be constructed by sorting and grouping all the bisector vertices. The details of determining and finalizing bisector vertices are described in Section 4.2 and 4.3 respectively.

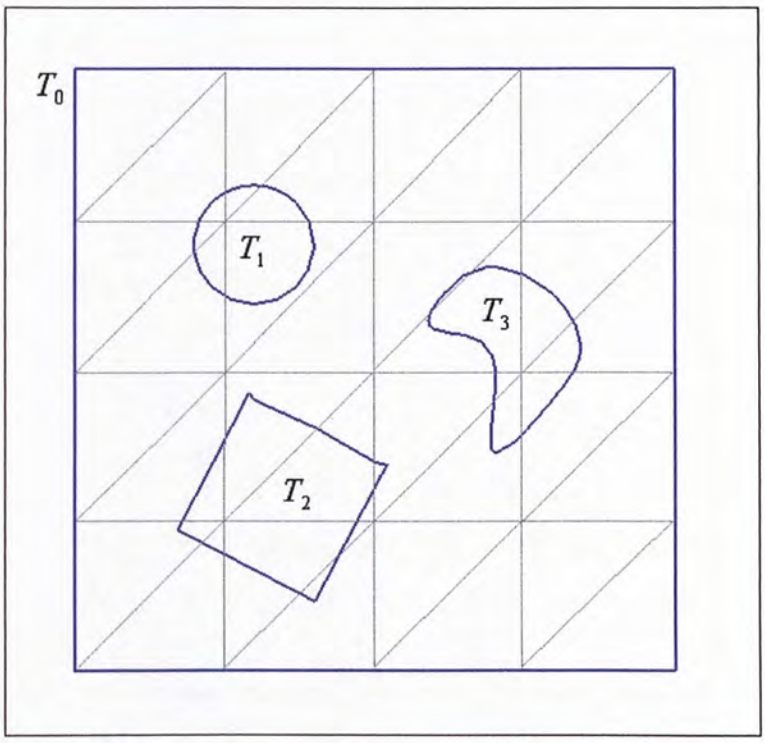

Figure 4.1 Triangulation of degree 4

4.1.1. Degree of Triangulation

For the triangulation shown in *Figure 4.1*, there are four pairs of triangular tiles in every row and column. It is referred to as a triangulation of degree 4. Triangulation of higher degree contains more triangular tiles, and hence more bisector vertices can be derived on the edges of the triangular tiles. This gives a more precise estimation of the bisectors. *Figure 4.2* shows the effect of the triangulation degree. The bisectors in *Figure 4.2a* are developed with lower degree triangulation while those in *Figure 4.2b* are developed with higher degree. It can be noticed that the bisectors bisect the valid region more accurately in *Figure 4.2b*. These bisectors will be refined in later steps so that the bisector vertices are evenly distributed on the bisectors (*Figure 3.3*).

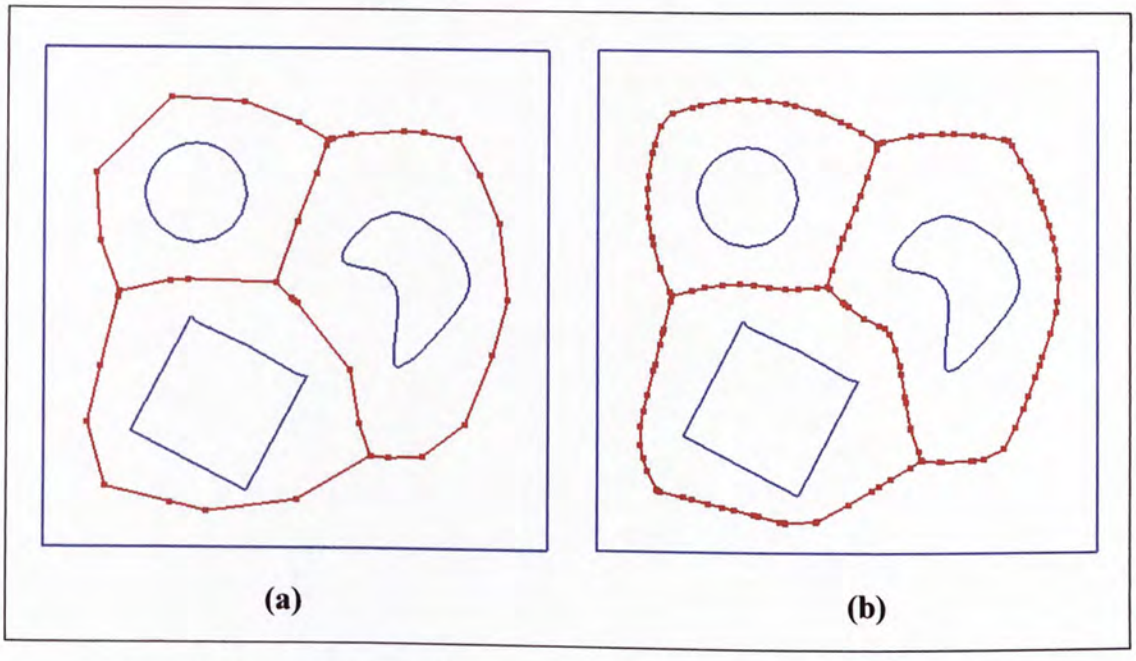

Figure 4.2 Voronoï diagrams by triangulations with different degrees

4.2. Locating Bisectors

After the parametric space is triangulated, the trims nearest to each vertices of every triangular tile are determined. Each vertex is then labelled with the index of its nearest trim. That is, if T_i is the nearest trim to a tile vertex, then the label of that vertex is *i*. To find the nearest trim of a tile vertex, the distances between the vertex and every trim are calculated by the method of point projection (*Figure 4.3*) [10]. The trim with the shortest distance to the vertex thus can be identified. Its index becomes the label of the vertex. In *Figure 4.4*, the numbers on the vertices of the triangular tile are their labels. There are four trims in the example so that the labels are 0, 1, 2 or 3.

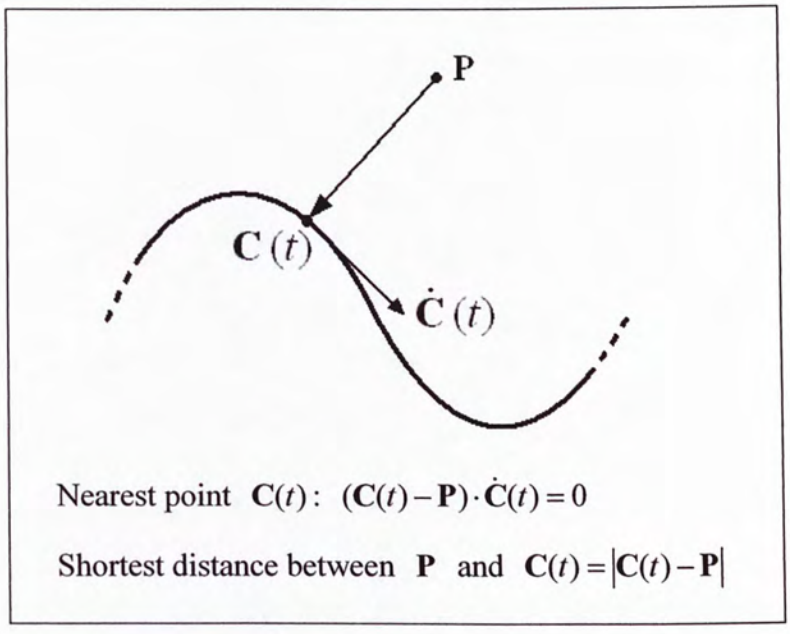

Figure 4.3 Point projection method

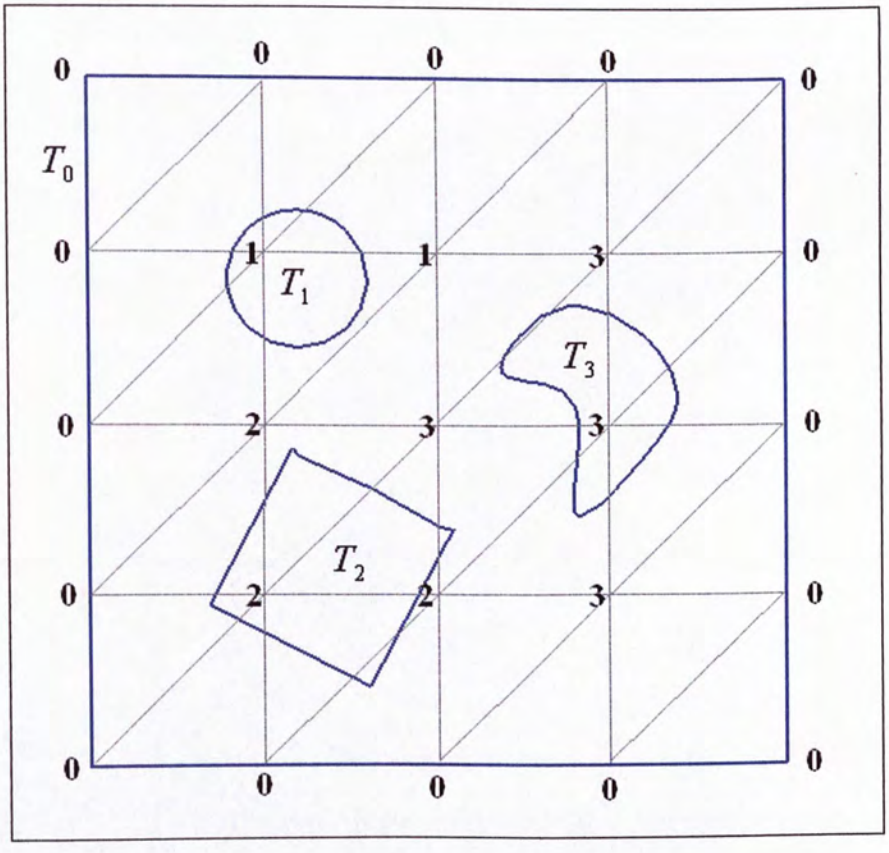

Figure 4.4 Labelling of triangular tiles

Among the three vertices of every single triangular tile, their labels can be the same or different. A vertex can be classified into three different types according to the label values. Different types of the vertices indicate the presence of bisector vertices on the edges of a triangular tile. The three cases are:

1. All three labels of the vertices are the same.
2. Two labels are different among the three of them.
3. All three labels of the vertices are different.

For case 1, the vertices are all nearest to the same trim. It implies that there is no bisector vertex exists on the edges of the triangular tile. No further process is necessary to be carried out on the triangular tile. An example is shown in *Figure 4.5*. The labels of the highlighted triangular tile are all equal to 0. Therefore, there is no bisector vertex on any of its edges.

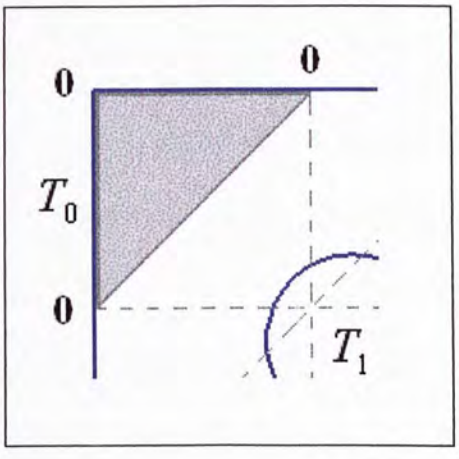

Figure 4.5 Same labels (case 1)

In case 2, two bisector vertices will be determined on the two edges with different labels at their end points. The bisector vertices are located at equal distance between the two nearest trims indicated by the labels. *Figure 4.6* is an example of

case 2. There are two different labels among the three labels of the highlighted triangular tile in the figure. They are 0 and 1. On each edge with different labels at its end points, there will be a bisector vertex equidistant from T_0 and T_1. They are represented by the two red dots in *Figure 4.6*. The red line joining the vertices is a segment of the bisector which bisects the valid region between T_0 and T_1.

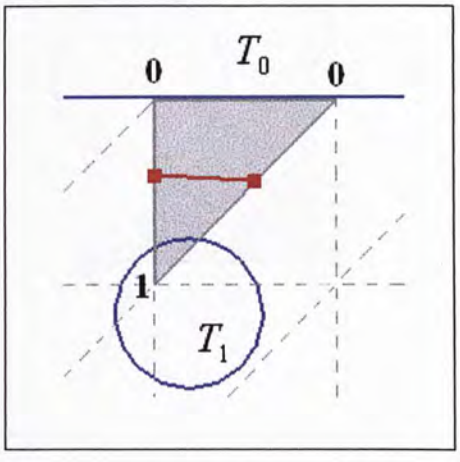

Figure 4.6 Two different labels (case 2)

Finally, for case 3, three different labels on the tile means that there are three bisectors intersecting the three edges of the triangular tile. Besides, these three bisectors meet at one point inside the tile. Therefore, four bisector vertices can be found in this case. Three of them are found on the edges of the tile. An extra one is found inside the tile. For example, in *Figure 4.7*, the labels at the vertices of the highlighted tile are 1, 2 and 3. Three bisector vertices are found on the edges and another one is found inside the tile. The three bisector vertices on the edges are determined by the same method as described in case 2. The bisector vertex inside the tile is called bisector centroid. A bisector centroid is determined by the property that it is equidistant from the nearest three trims. The three trims are indicated by the labels of the tile vertices. In *Figure 4.7*, the three nearest trims to the bisector centroid are

T_1, T_2 and T_3. Thus, the bisector centroid is of equal distance from T_1, T_2 and T_3. These four determined bisector vertices form three segments of the three bisectors meeting at the bisector centroid. In the figure, they bisect the valid regions between T_1 and T_2, T_2 and T_3, and T_3 and T_1 respectively.

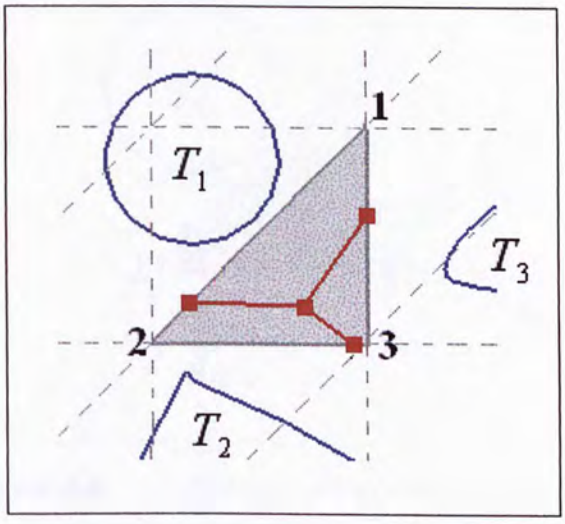

Figure 4.7 Three different labels (case 3)

4.2.1. Bisector Centroids

A bisector vertex on an edge of a triangular tile can be easily located by using numerical methods such as the method of bisection. When case 3 is encountered, locating the bisector centroid within the area of the triangular tile involves recursive numerical computation. However, the computation of bisector centroid is more complicated.

To locate the bisector centroid, the bisector vertices on the three edges of the triangle have to be located. Then an initial point, denoted by $\mathbf{c}_0$, is obtained by taking the average of the coordinates of the bisector vertices (*Figure 4.8*). The shortest distances between $\mathbf{c}_0$ and the three nearest trims are calculated. If these distances are all equal, or the greatest difference between them is less than a prescribed tolerance,

the point will be considered as the bisector centroid and thus no further recursive computation is required. Otherwise, if this condition is not satisfied, the recursion begins.

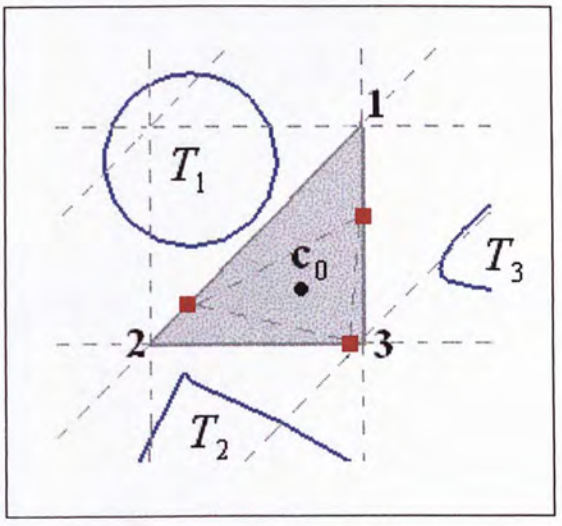

Figure 4.8 Average of bisector vertices ($\mathbf{c}_0$)

On each of the three nearest trims, a point which is nearest to $\mathbf{c}_0$ is located. Let these three points be $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$. As illustrated in *Figure 4.9*, the blue dots are the locations of the trim points nearest to $\mathbf{c}_0$. The center of the circle passing through $\mathbf{r}_1$, $\mathbf{r}_2$ and $\mathbf{r}_3$ is calculated (*Figure 4.10*). Denoting the circle center by $\mathbf{c}_1$, its distances between the three nearest trims are determined. If the distances are equal or with difference less than the tolerance, the terminating condition is satisfied and $\mathbf{c}_1$ will be taken as the bisector centroid. If not, $\mathbf{c}_1$ will replace $\mathbf{c}_0$ and the process is repeated for $\mathbf{c}_i$, $i = 0, 1, 2...$, until the terminating condition is satisfied. The final result of the recursion is a bisector centroid located at equal distance from the three nearest trims.

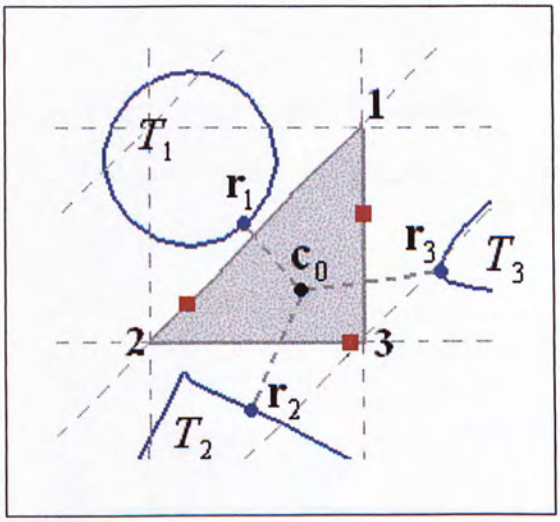

Figure 4.9 Nearest points on the trims

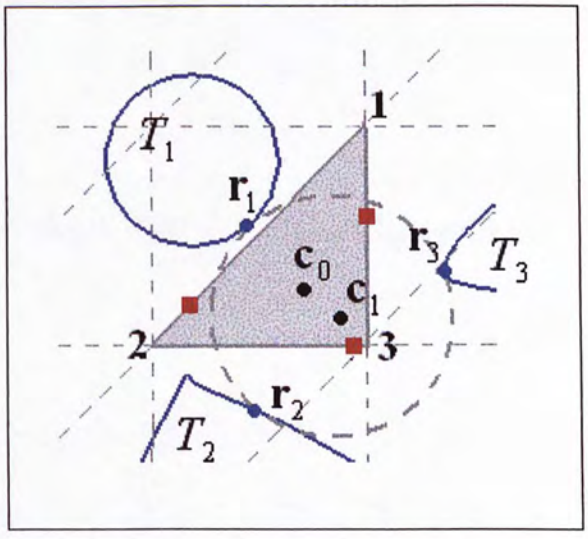

Figure 4.10 Center of circle passing through the nearest trim points ($\mathbf{c}_1$)

4.2.2. Sub-triangulation

In the process of locating bisector vertices on the edges of a triangular tile, it is possible that a bisector vertex is closer to a trim other than the two trims indicated by the labels at the ends of the edge. *Figure 4.11* shows an example of such case. The labels at the end points of the diagonal of the highlighted tile are 2 and 3. A bisector

vertex between T_2 and T_3 is thus determined on the edge. However, this bisector vertex, i.e., the red dot in the figure, is found to be nearer to T_1 rather than T_2 and T_3. This implies that the degree of initial triangulation is not high enough to derive the bisector vertices for the trims.

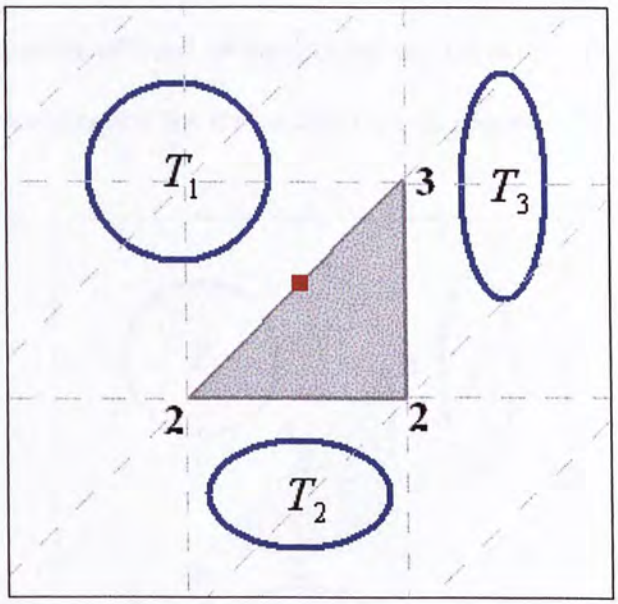

Figure 4.11 Special bisector vertex

Sub-triangulation means that the triangular tile encountering such special case is subdivided into four triangular tiles. The subdivision is applied at the computed bisector vertices or mid points of the three edges of the tile. For example, in *Figure 4.12*, the horizontal edge of the highlighted tile is labelled by the index 2 at both of its end points. Then the mid point of this edge is determined. Assume this mid point's nearest trim is T_2 and thus it is also labelled by 2. For the vertical edge in the figure, its end points are labelled as 2 and 3. Assume the nearest trim of the bisector vertex determined on this edge be T_2 or T_3. Therefore, the label of this bisector vertex is 2 or 3. Together with the special bisector vertex labelled by 1 on the diagonal edge, subdivision is carried out on the lines joining these three points. The original tile is

subdivided into four triangular tiles, as indicated by the grey lines across the highlighted tile. The determination of bisector vertices continues to be carried out on each of these four tiles according to the labels at their vertices. Sub-triangulation will be recursively applied on these tiles whenever there is a special bisector vertex. However, if the longest edge of the subdivided tile is shorter than a prescribed tolerance, the subdivision will not be carried out on that small tile. *Figure 4.13* is the result of bisector determination for the special case in *Figure 4.12*.

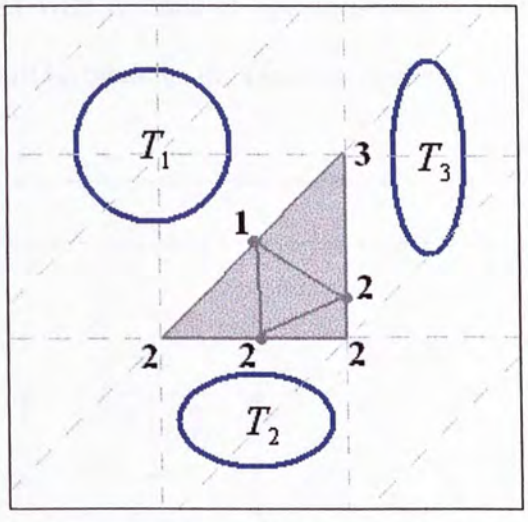

Figure 4.12 Sub-triangulation

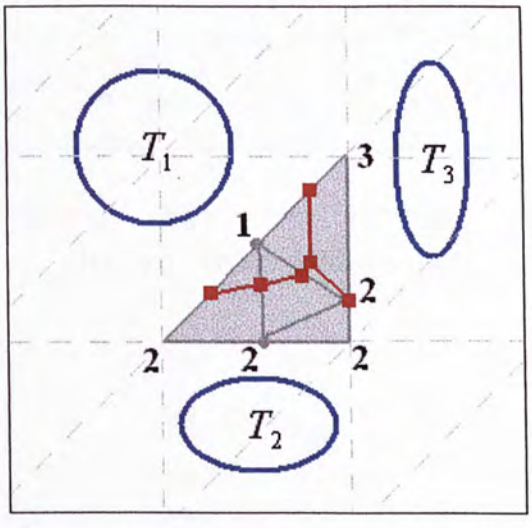

Figure 4.13 Bisector vertices and bisector centroid by sub-triangulation

4.3. Finalizing Bisectors

After all the line segments of the bisectors are determined by triangulation, they are ready to be grouped and finalized. First, the line segments in the triangular tiles are connected at the common bisector vertices. *Figure 4.14* shows the line segments connected to form the bisectors. This results in every trim being bounded by a closed loop of bisector. However, the bisector vertices on the bisector loops are not evenly distributed. In order to have an even distribution of bisector vertices, the bisector vertices are interpolated with a cubic B-Spline curve. Uniformly distributed points sampled on the curve will be taken as the bisector vertices.

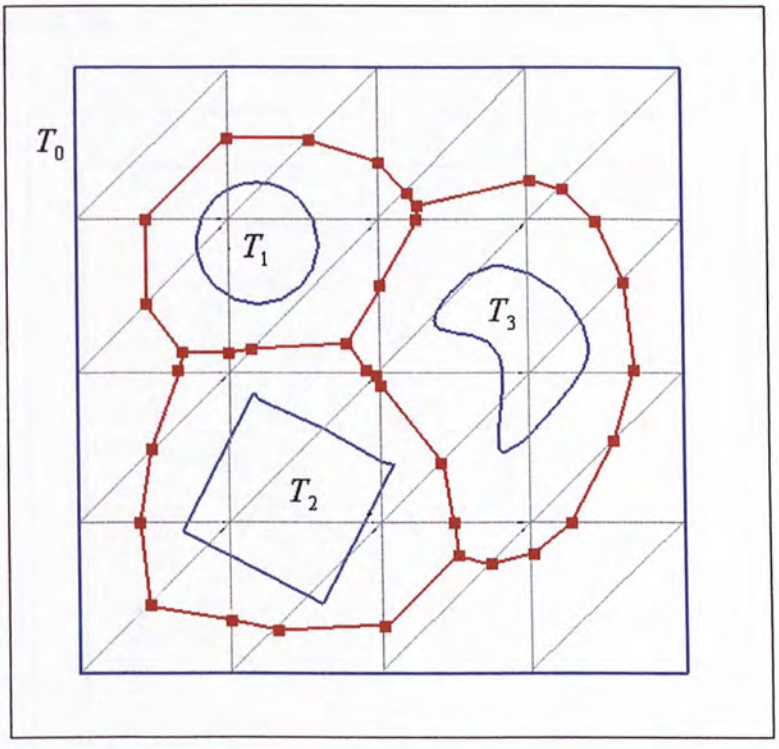

Figure 4.14 Bisector vertices joined to form closed bisectors

Because of the discontinuity at the bisector centroids, the closed bisector loops are divided at the bisector centroids. The segments between the bisector centroids are interpolated with a cubic B-Spline. *Figure 4.15* is the interpolated curves between the bisector centriods. In order to preserve the shape of the bisector, the first derivatives at the ends of the curve segments are specified [10]. The final step is to sample data points on the curves. The number of sample points on every curve should be chosen such that the bisector vertices sampled on all curves are uniformly distributed. These points are the finalized bisector vertices. The number of sample points also affects the later stages such as correspondence establishment and surface approximation, and will be discussed in later sections. The sampling result of the model in *Figure 4.15* is shown in *Figure 4.16*.

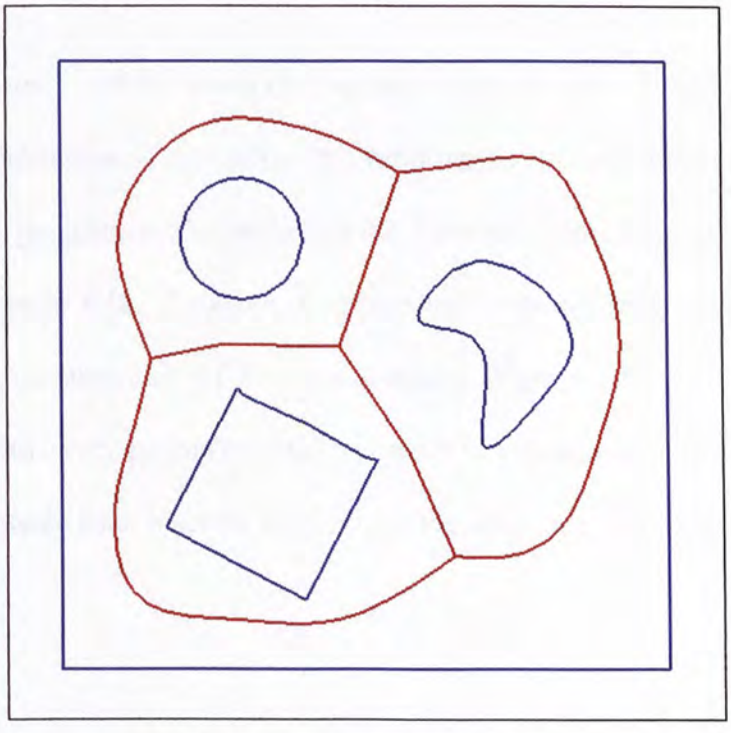

Figure 4.15 Curve fitting of bisector vertices between bisector centroids

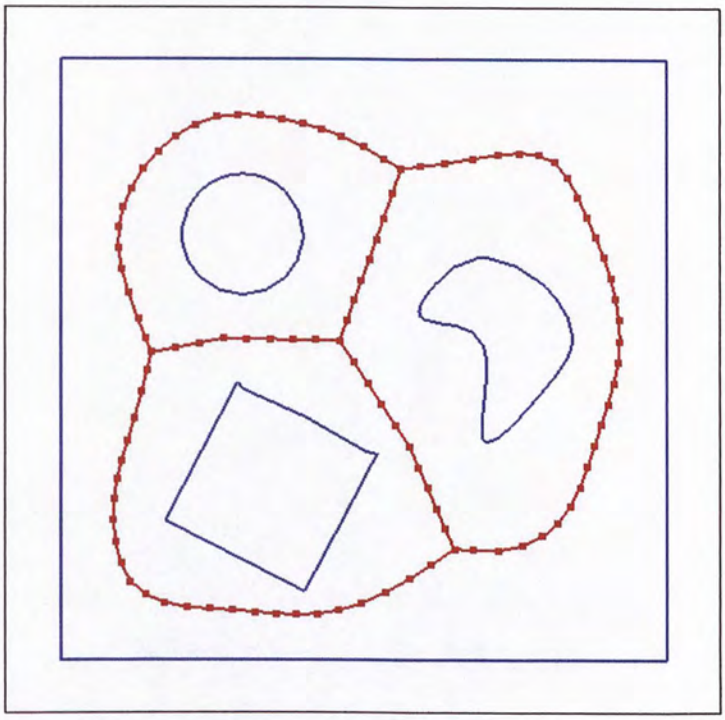

Figure 4.16 Sample points on the bisector curve

The construction of the Voronoï diagram results in a set of closed bisector loop defining the valid region of each trim. The valid region between a trim and its bisector loop is called a parametric tile. Including the boundary trim, there are four trims for the model in *Figure 4.16*. Therefore, four bisector loops and four parametric tiles are derived by the development of Voronoï diagram. *Figure 4.17* is an illustration of *Figure 4.16*, with every parametric tile separated to indicate their existence. There is only one trim inside each bisector loop except the boundary trim, which is outside its bisector loop.

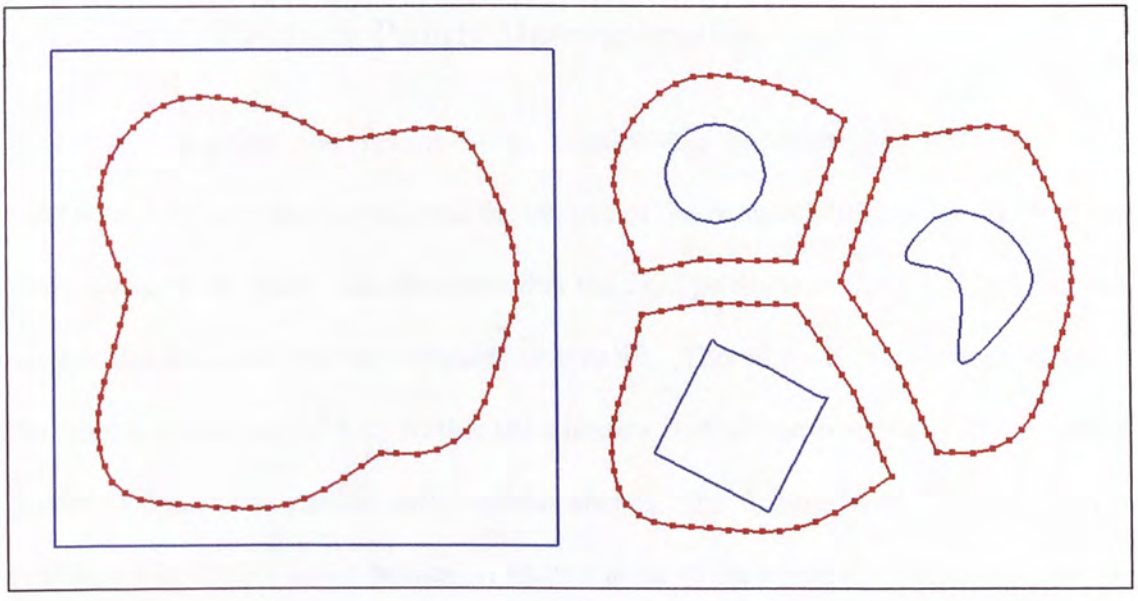

Figure 4.17 Parametric tiles

According to the nature of a bisector, the shape of a trim should be roughly similar to the shape of its bisector loop, although the shape of its bisector maybe affected by the shapes of its neighbouring trims. A trimmed surface is decomposed by considering the shapes and features of the trims and their bisectors.

Chapter 5. Feature Points Determination

Before constructing the feature-based relationship between the trims and their bisectors, it is necessary to analyse the shapes of the trims. Feature points refer to the sharp turns on the trims. It is desirable that the final products of the algorithm, i.e., the surface patches, are shaped as regular as possible. Therefore, it is important to locate the feature points on the trim so that the trimmed surface can be divided at the feature points to obtain the patches with regular shapes. The feature point determination is performed in the 3D space because a feature point in the parametric space may not be a feature point on the physical surface. Since the surface patches are considered, the feature points are determined in the 3D space rather than the parametric space.

5.1. Definitions of Feature Points

There are two different kinds of feature points which can be found on the trims. They are classified into two types according to their positions and geometric properties. The first type is known as continuous sharp turns and the other is called discrete sharp turns. Continuous sharp turns are found on the trimming curves composing a trim. To identify whether a trimming curve data point is a sharp turn, the ratio between the radius of curvature of the data point and the total arc length of the trim to which the trimming curve belongs is evaluated [9]. If the ratio is less than a certain threshold, the data point is defined as a continuous sharp turns. Unlike continuous sharp turns, discrete sharp turns are found at the junctions between the trimming curves of a trim. The interior angle of a junction is measured to test if it is small enough to be classified as a discrete sharp turn.

5.1.1. Continuous Sharp Turns

The sharp turns found on a continuous trimming curves of the trims are called continuous sharp turns. A data point on a trimming curve is regarded as a sharp turn if the ratio between its radius of curvature and the total arc length of the trim is smaller than a threshold. To derive a value for the threshold, the case of a data point on a unit circle is considered. For any point on a unit circle, its radius of curvature is 1 unit and the total arc length of the unit circle is 2π. Therefore, the ratio ε between the radius of the data point and the total arc length is $\frac{1}{2\pi}$. This sharpness ratio is used as the threshold for defining whether a curve data point is a sharp turn. In *Figure 5.1*, the difference between a sharp turn and a smooth turn is illustrated. The data point on the unit circle has a sharpness ratio of $\frac{1}{2\pi}$. The data point on the deformed part of the unit circle has a radius of curvature less than 1 unit and the total arc length of the deformed circle is larger than 2π. Therefore, its sharpness ratio is less than $\frac{1}{2\pi}$ and is classified as a sharp turn, i.e., a feature point on the deformed circle.

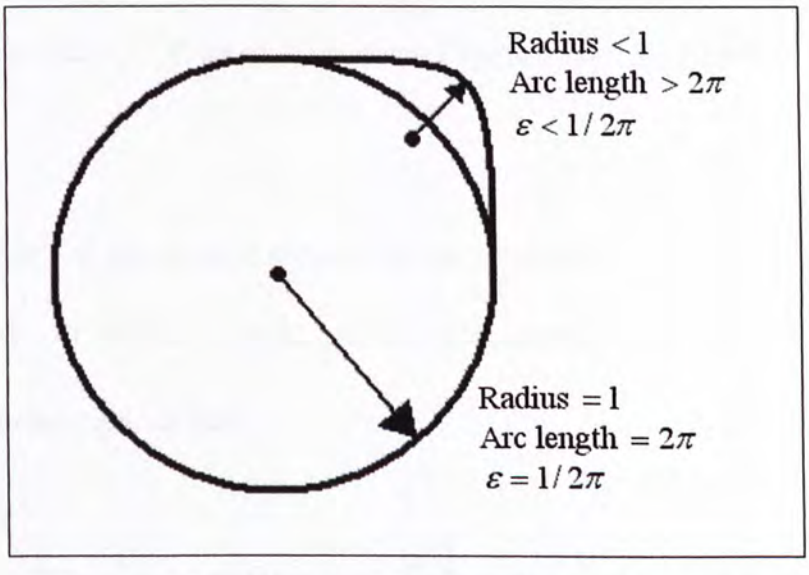

Figure 5.1 Definition of continuous sharp turn

To identify continuous sharp turns on a trimming curve, the location of possible sharp turns on a curve must be estimated. This is performed by projecting the control vertices onto the curve [10]. It is because the shape and features of a curve is controlled by its control vertices. After the projection, a set of approximate trimming curve segments are obtained. Two further steps will be carried out on the projected control vertices to decide whether they are sharp turns and to locate the position of shape turns if they exist.

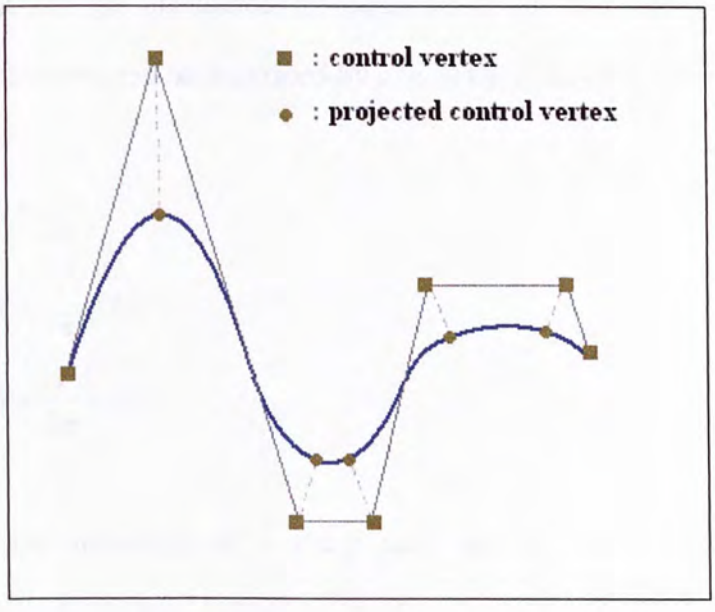

Figure 5.2 Projecting the control vertices onto the trimming curve

In *Figure 5.2*, the control vertices are projected on a trimming curve. Given a data point $\mathbf{C}_{i,j}(t)$ which is on the j-th trimming curve of the trim T_i, its radius of curvature can be expressed as

$$\rho_{i,j}(t) = \frac{\left|\dot{\mathbf{C}}_{i,j}(t)\right|^3}{\left|\dot{\mathbf{C}}_{i,j}(t) \times \ddot{\mathbf{C}}_{i,j}(t)\right|} \tag{5.1}$$

where $\dot{\mathbf{C}}_{i,j}(t)$ and $\ddot{\mathbf{C}}_{i,j}(t)$ denote the first and second derivatives of $\mathbf{C}_{i,j}(t)$ respectively [11]. Assuming the total arc length of the trim T_i is $L(T_i)$, the sharpness ratio of $\mathbf{C}_{i,j}(t)$ is then

$$\varepsilon_{i,j}(t) = \frac{\rho_{i,j}(t)}{L(T_i)} \tag{5.2}$$

Since the projected control vertices are also data points on the curve, their sharpness ratios can be calculated by equation (5.2). The sharpness ratio of a projected control vertex can be described by one of the following three cases:

1. $\varepsilon_{i,j}(t) < \frac{1}{2\pi}$
2. $\varepsilon_{i,j}(t) \le \frac{1}{2\pi} + \tau_{i,j}$
3. $\varepsilon_{i,j}(t) > \frac{1}{2\pi} + \tau_{i,j}$

where $\frac{1}{2\pi}$ is the threshold of a sharp turn and $\tau_{i,j}$ is a predefined positive tolerance. Case 1 indicates that the projected control vertex is a sharp turn. For case 2, although the vertex is not a sharp turn, it is possible that a sharp turn exists in its vicinity as shown in *Figure 5.3*. Assume the two projected control vertices in the figure are both with sharpness ratio larger than $\frac{1}{2\pi}$ but smaller than $\frac{1}{2\pi} + \tau_{i,j}$. By the nature of the control polygon, there can be a sharper data point between or near these two projected control vertices. It means that this data point's sharpness ratio is at least smaller than $\frac{1}{2\pi} + \tau_{i,j}$. It also implies that the sharpness ratio of the data point maybe even smaller than $\frac{1}{2\pi}$. In this case, the data point between that two projected control vertices is defined as a sharp turn. In case 3, for those projected control

vertices with sharpness ratios larger than $\frac{1}{2\pi}+\tau_{i,j}$, they are not considered as sharp turns and there is no sharp turn in its vicinity. Hence, the projected control vertices whose sharpness ratios satisfying case 1 and 2 will be processed further for the identification of continuous sharp turns.

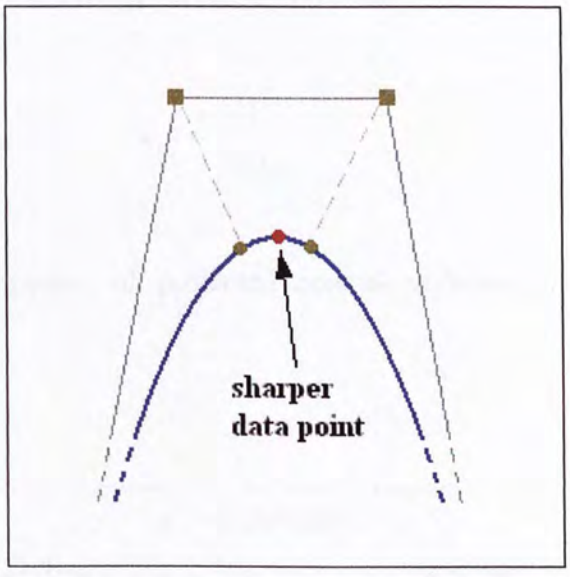

Figure 5.3 Data point sharper than the projected control vertices

The value of the tolerance $\tau_{i,j}$ depends on the number of the projected control vertices on the trimming curve $\mathbf{C}_{i,j}(t)$. The more the number of projected control vertices on a trimming curve, the smaller is the tolerance. This is because the approximate curve segments estimated by the projected control vertices is more accurate if the number of the projected control vertices is larger. As a comparison in *Figure 5.4*, the two trimming curves are of the same shape but with different number of projected control vertices. Sharp turns on the one with more projected control vertices can be easily located since some of the control vertices are projected nearer to the sharp turns (*Figure 5.4b*). Therefore, it is unnecessary to set the tolerance $\tau_{i,j}$ to

a large value. Otherwise, some projected control vertices which are not so close to the sharp turns maybe classified as location of sharp turn. Thus, fewer vertices will be processed in the following step and the computation effort is reduced. One way to define $\tau_{i,j}$ is to divide the sharpness ratio threshold by the number of projected control vertices on the trimming curve $\mathbf{C}_{i,j}(t)$, i.e.,

$$\tau_{i,j} = \frac{1}{2\pi n_{i,j}} \tag{5.3}$$

where $n_{i,j}$ is the number of projected control vertices on the trimming curve $\mathbf{C}_{i,j}(t)$.

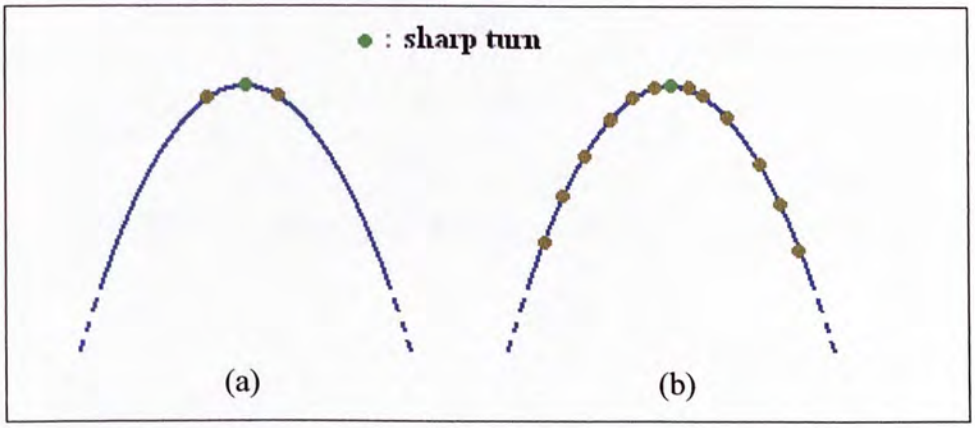

Figure 5.4 Trimming curves with different number of projected control vertices

The above process gives a set of possible sharp turns on the curve. The next step is to determine a more precise location for each of the sharp turns. For those projected control vertices which satisfy case 1 in the previous test, they are considered as the curve's sharp turns and no further process will be performed. For those satisfying case 2, each pair of adjacent projected control vertices defines a range containing the sharp

turns. Sample points on the curve are generated by dividing this range evenly. The data point with sharpness ratio less than $\frac{1}{2\pi}$ will be considered as the sharp turn. The result of locating sharp turns for the example in *Figure 5.2* is shown in *Figure 5.5*. It consists of all the three cases. Two continuous sharp turns are determined.

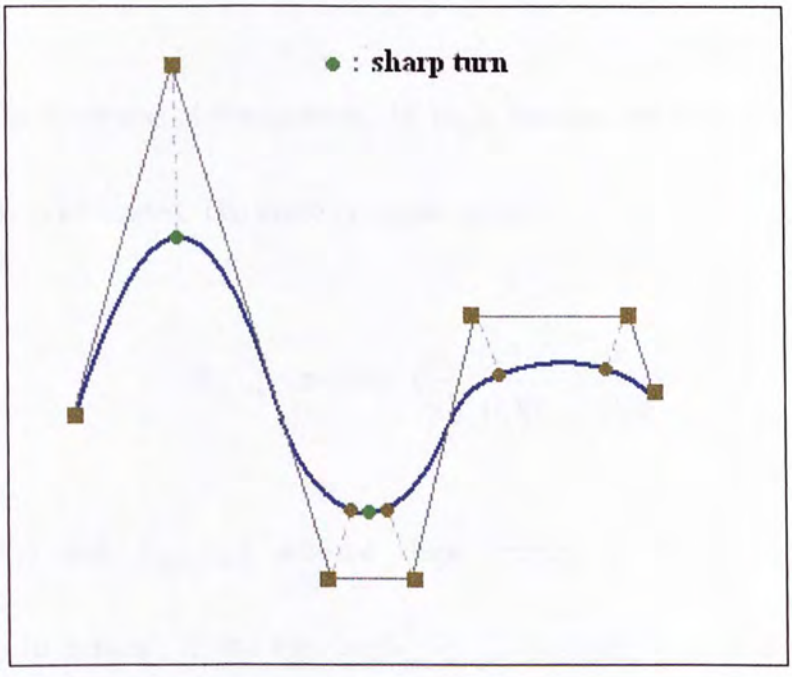

Figure 5.5 Sharp turns determined for the example in Figure 5.2

5.1.2. Discrete Sharp Turns

Since a trim can be composed of more than one trimming curves, sharp turns may exist at the junctions between any two connected trimming curves. This kind of sharp turns is known as discrete sharp turns. Instead of making use of the sharpness ratio, the interior angle at the junction is measured to define whether the junction is a sharp turn. The interior angle is the turn angle between the slope vectors of the two curves

at the junction. Consider the junction of the curves $\mathbf{C}_{i,j}(t)$ and $\mathbf{C}_{i,j+1}(t)$, $t \in [t_0, t_1]$ on the trim T_i. In *Figure 5.6*, assuming the curves are C^0 continuous, we have

$$\begin{aligned} \mathbf{C}_{i,j}(t_1) &= (x(t_1), y(t_1), z(t_1)) \\ &= \mathbf{C}_{i,j+1}(t_0) = (x(t_0), y(t_0), z(t_0)) \end{aligned} \tag{5.4}$$

To identify the sharpness of the junction, the angle between the derivatives of $\mathbf{C}_{i,j}(t)$ and $\mathbf{C}_{i,j+1}(t)$ is evaluated. The angle is expressed as

$$\theta_{i,j,j+1} = \pi - \cos^{-1}\left(\frac{\dot{\mathbf{C}}_{i,j}(t_1) \cdot \dot{\mathbf{C}}_{i,j+1}(t_0)}{\left|\dot{\mathbf{C}}_{i,j}(t_1)\right| \left|\dot{\mathbf{C}}_{i,j+1}(t_0)\right|}\right) \tag{5.5}$$

where $\dot{\mathbf{C}}_{i,j}(t_1)$ and $\dot{\mathbf{C}}_{i,j+1}(t_0)$ are the slope vectors of $\mathbf{C}_{i,j}(t_1)$ and $\mathbf{C}_{i,j+1}(t_0)$ respectively. In general, if the turn angle $\theta_{i,j,j+1}$ is equal to or less than $\frac{\pi}{2}$, the junction is regarded as a sharp turn.

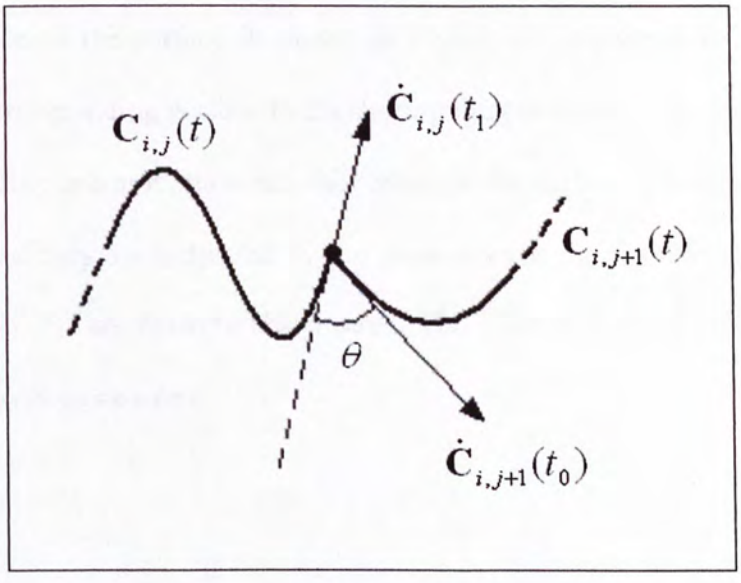

Figure 5.6 Definition of discrete sharp turn

5.2. Parametric Coordinates of Feature Points

Up to this stage, all the feature points, including the continuous and discrete sharp turns, are determined. And the feature points are expressed by their coordinates on the physical surface. The correspondence between the feature points and the bisector vertices will have to be established. Since this has to be performed in the parametric space, the parametric coordinates of the feature points must be known. To calculate the feature points' parametric coordinates, the method of point inversion is used [10]. By specifying the coordinate of a curve or surface data point, the parametric value of the data point can be determined by the point inversion method. Hence, the *j*-th feature point on the *i*-th trim T_i, it can be expressed as

$$\mathbf{P}_{i,j}^{\mathrm{f}} = \mathbf{S}(u_{i,j}^{\mathrm{f}}, v_{i,j}^{\mathrm{f}}) = (x_{i,j}^{\mathrm{f}}, y_{i,j}^{\mathrm{f}}, z_{i,j}^{\mathrm{f}}) \tag{5.6}$$

where $(u_{i,j}^{\mathrm{f}}, v_{i,j}^{\mathrm{f}})$ is the parametric coordinate and $(x_{i,j}^{\mathrm{f}}, y_{i,j}^{\mathrm{f}}, z_{i,j}^{\mathrm{f}})$ is the Euclidean coordinate of the feature point.

Figure 5.7 and *Figure 5.8* are examples of feature points determination. The parametric space of the surface is shown in *Figure 5.7*. *Figure 5.8* shows different views of the corresponding surface to illustrating the positions of the feature points on the surface. In this example, there are four trims on the surface. There are nine feature points found and they are indicated by the green dots in the figures. The four feature points found on T_0 are discrete sharp turns. The other five feature points found on T_1 are continuous sharp turns.

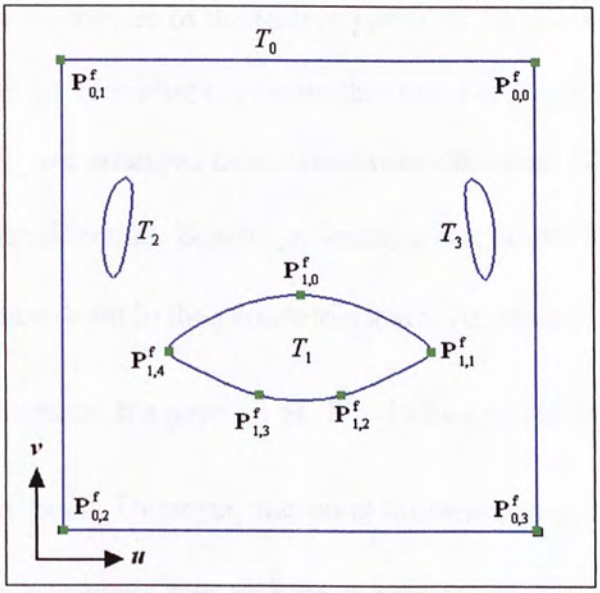

Figure 5.7 Feature points in parametric space

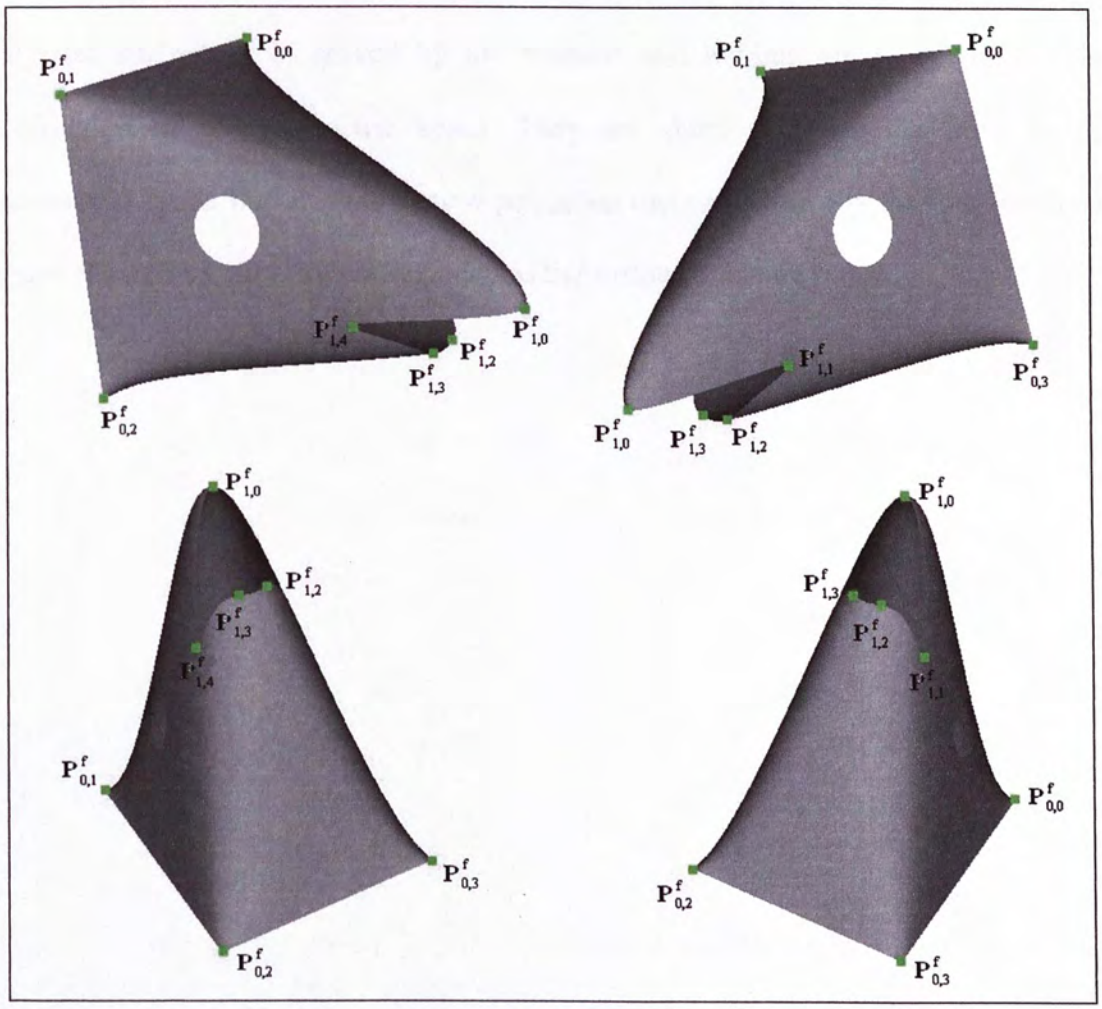

Figure 5.8 Feature points on physical surface

In the example, the indices of the feature points on the same trim are arranged in the same direction as the trimming curves on the trim. For example, the indices of the feature points on T_0 are arranged in anticlockwise direction while those on T_1 are arranged in clockwise direction. Besides, a feature point on the physical surface does not have to be a feature point in the parametric space. An obvious example is $\mathbf{P}_{1,0}^{f}$ on T_1. On the physical surface, the position of $\mathbf{P}_{1,0}^{f}$ is the apex of the model and hence it is one of the feature point. However, the point corresponding to $\mathbf{P}_{1,0}^{f}$ on T_1 in the parametric space is a smooth turn and its sharpness ratio is obviously not small enough to be defined as a feature point. The same situation occurs on $\mathbf{P}_{1,2}^{f}$ and $\mathbf{P}_{1,3}^{f}$. Conversely, a feature point in the parametric space may not be a feature point on the physical surface. It is proved by the maxima and minima on T_2 and T_3 in the v-direction of the parametric space. They are sharp turns on the trims in the parametric space but are not feature points on the surface in the three-dimensional space. Therefore, they are not regarded as the surface's feature points.

Chapter 6. Vertices Correspondence Establishment

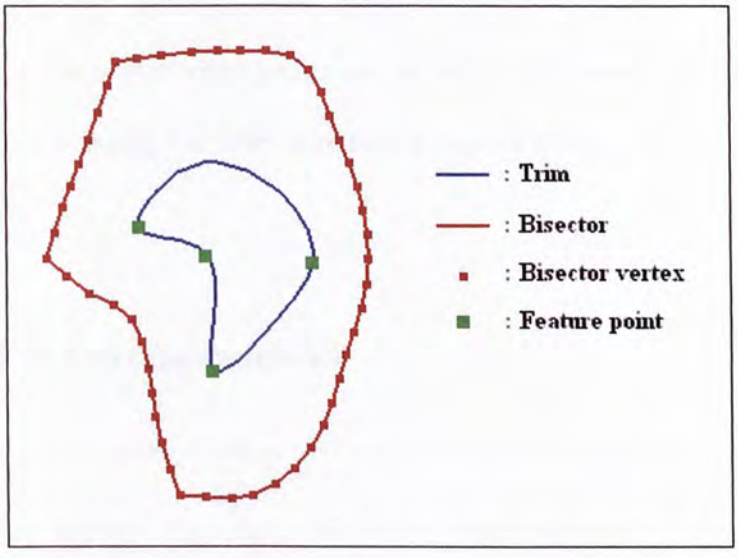

Figure 6.1 Data in a parametric tile

After the Voronoï Diagram is developed for the surface in the parametric space, every trim has its own isolated valid region defined by a bisector loop. This isolated valid region is referred to as a parametric tile. Feature points are then determined on every trim. With all these data prepared in every parametric tile (*Figure 6.1*), correspondences will be established between the feature points and the bisector vertices. By considering the similarity between the feature points and the bisector vertices, every feature point on a trim is linked to the bisector vertex with the highest degree of similarity. A shape blending approach is adopted to measure the similarity between the feature points and the bisector vertices. The process starts by normalizing the shapes of the trim and its bisector loop so that both of them are enclosed by a unit square. A ranking process is performed on the vertices of the normalized shapes. The interior angles of the bisector vertices and the normalized distances between the feature points and the bisector vertices in the 3D space are considered in the ranking

process. The result of the process is that a feature point on the trim is associated with the bisector vertex with the highest ranking. In most case, there are sharp interior angles at the bisector centroid where three bisectors meet. Bisector centroids are associated with their nearest data points on the trims. The parametric tile will later be divided into patches along the lines connecting feature points and their corresponding bisector vertices.

6.1. Validity of Correspondences

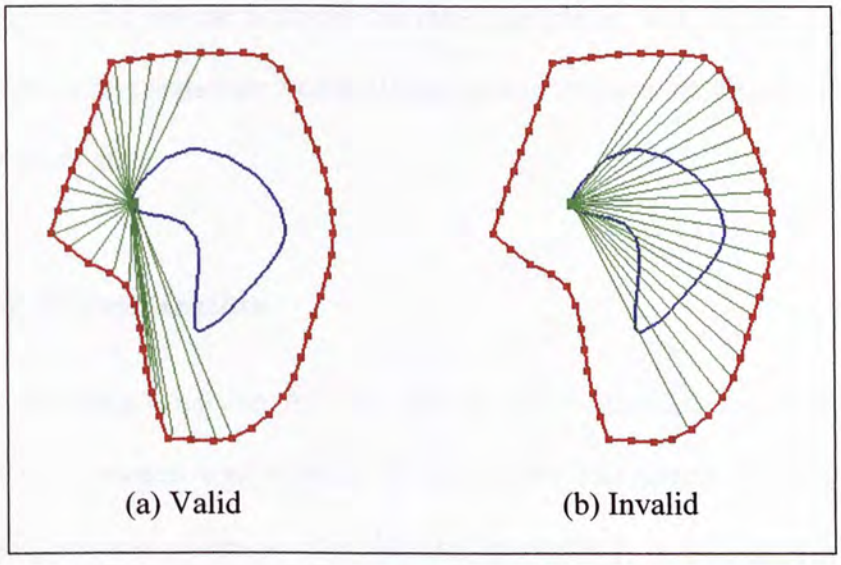

Figure 6.2 Correspondence links

A valid correspondence means that the straight line joining a feature point and a bisector vertex does not intersect the trim at points other than the feature point. It is because these straight lines partition the trimmed surface into patches. These straight lines should lie completely inside the valid region of the parametric tile. This straight line which represents the correspondence is called correspondence link, or link in

short. Only those bisector vertices with valid correspondences will be ranked in the ranking process.

In *Figure 6.2*, lines are constructed connecting a trim's feature point to all the bisector vertices to test the validity of the correspondence. In *Figure 6.2b*, the correspondences of the connected bisector vertices are invalid as their correspondence links intersect the trim not only at the feature point. In *Figure 6.2a*, the correspondences of the connected bisector vertices are valid. They are the possible choices for associating with the feature point. The same identification process is carried out on all the feature points of the other parametric tiles. In this way, only the possible valid correspondences within all parametric tiles are identified and processed in the subsequent steps.

6.2. Shape Normalization

The shape blending approach by Hui and Li [9] is adopted for establishing the correspondence between a trim and a bisector loop. The shapes of the trim and its bisector loop are normalized so that they are enclosed in a unit square. A ranking process is carried out so that a feature point is associate with the nearest and sharpest bisector vertex in the normalized domain.

In *Figure 6.3*, the angles of all the bisector vertices are approximately equal. *Figure 6.3a* shows the correspondence between the trim and the bisector without normalizing the trim and the bisector. The feature points are directly linked to the nearest bisector vertices, ignoring the shapes of the trim and the bisector loop. This results in the undesirable shapes of parametric patches. However, if normalization is performed before the ranking process, the feature points will be linked to the most similar vertex on the bisector loop, i.e., the nearest bisector vertices in the normalized

domain. This correspondence is established by considering the similarity between the shapes of the trim and its bisector loop (*Figure 6.3b*).

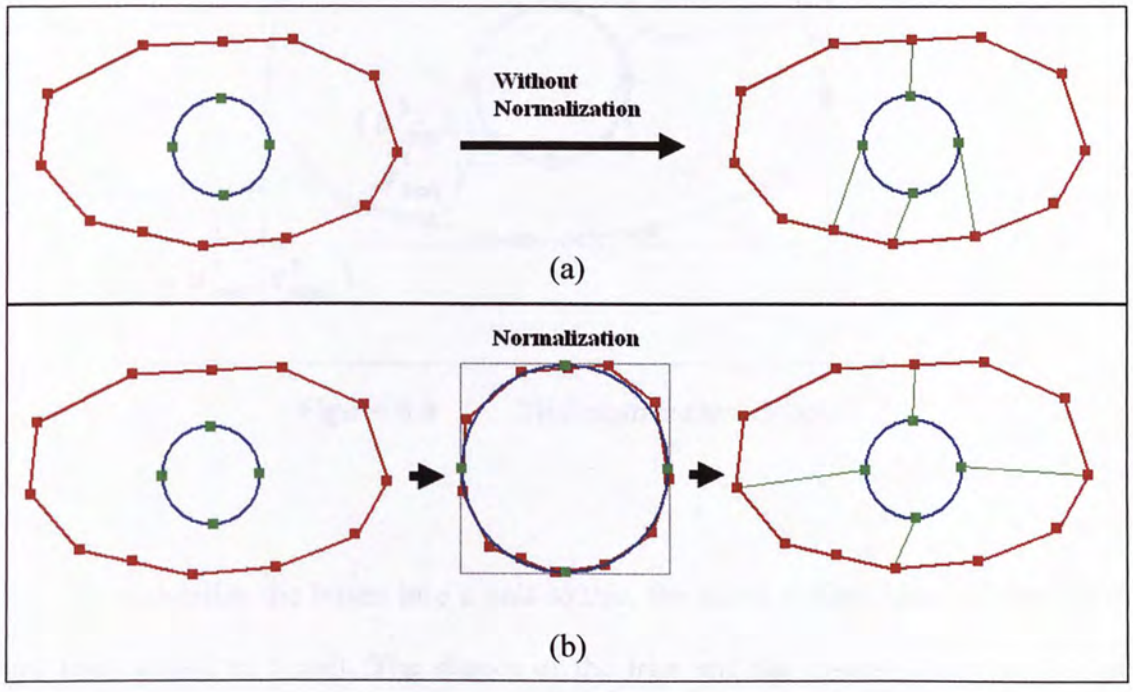

Figure 6.3 Effect of normalization

The normalization is performed by computing the minimum enclosing boxes for the trim and its bisector loop in the same parametric tile. These boxes, together with the enclosed shapes by them, are normalized into a unit square. For example, if the maximum and minimum *u*-coordinates of the trim are u^t_{max} and u^t_{min}, and the maximum and minimum *v*-coordinates are v^t_{max} and v^t_{min}, the minimum enclosing box for the trim will have diagonal coordinates (u^t_{min}, v^t_{min}) and (u^t_{max}, v^t_{max}). Similarly, the diagonal coordinates of the minimum enclosing box for the bisector loop is denoted by (u^b_{min}, v^b_{min}) and (u^b_{max}, v^b_{max}). A graphical interpretation is shown in *Figure 6.4*.

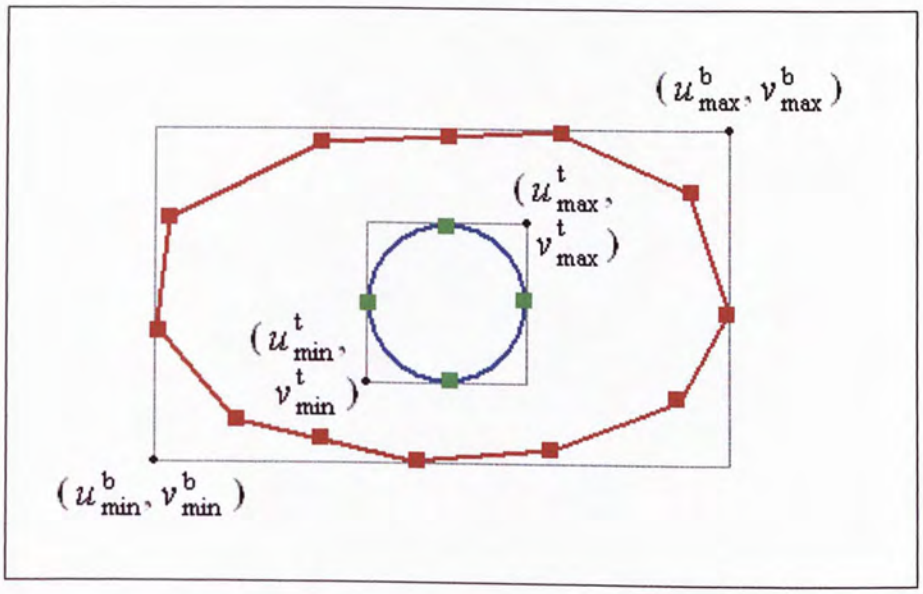

Figure 6.4 Minimum enclosing boxes

To normalize the boxes into a unit square, the u and v dimensions of the boxes are both scaled to 1 unit. The shapes of the trim and the bisector loop are scaled accordingly. Moreover, the normalized boxes, together with the enclosed shapes, are translated so that the normalized boxes of the trim and the bisector loop are aligned. Therefore, the normalized coordinates of a data point (u^{t}, v^{t}) on the trim can be expressed as

$$(u'^{t}, v'^{t}) = \left(\frac{u^{t} - u^{t}_{min}}{u^{t}_{max} - u^{t}_{min}}, \frac{v^{t} - v^{t}_{min}}{v^{t}_{max} - v^{t}_{min}}\right) \tag{6.1}$$

The normalized feature points on the trim can also be expressed in this way because they are also the data points on the trim. Similarly, the normalized coordinates of a bisector vertex (u^{b}, v^{b}) can be expressed as

$$(u'^{b}, v'^{b}) = \left(\frac{u^{b} - u^{b}_{min}}{u^{b}_{max} - u^{b}_{min}}, \frac{v^{b} - v^{b}_{min}}{v^{b}_{max} - v^{b}_{min}}\right) \tag{6.2}$$

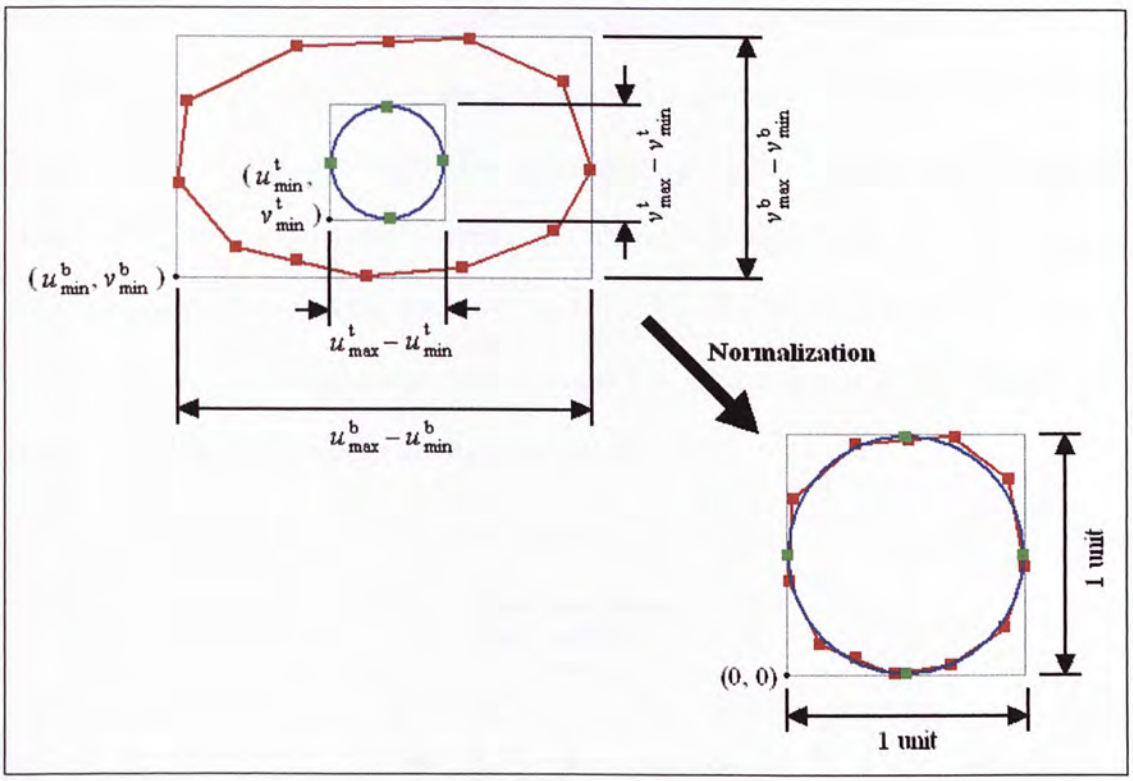

Figure 6.5 Normalization

An example of the normalized shapes of a trims and a bisector loop can be found in *Figure 6.5*. Note that the scale of a shape in the *u*- and *v*-directions maybe different and the enclosing boxes may completely overlap after normalization. For each parametric tile, all feature points on the trim and all valid bisector vertices are normalized using Equation (6.1) and Equation (6.2) respectively. Correspondence between the feature points and the bisectors can then be established based on the normalized shapes and the ranking process.

6.2.1. Normalization with Relative Position

In some cases, the positions of the original shapes of the trims and its corresponding bisector loop are different. If the difference is significant compared with the sizes of

the shapes, the shapes of the developed patches will be undesirable, even though normalization is performed. *Figure 6.6a* shows an example of the undesirable result. In the example, the centers of the trims and its bisector loop do not coincide. However, in the normalized domain, the shapes are aligned so that their centers coincide. After the ranking process, some of the feature points on the trim are associated with the bisector vertices that are not closest to the feature points. The shapes of the divided patches are thus undesirable.

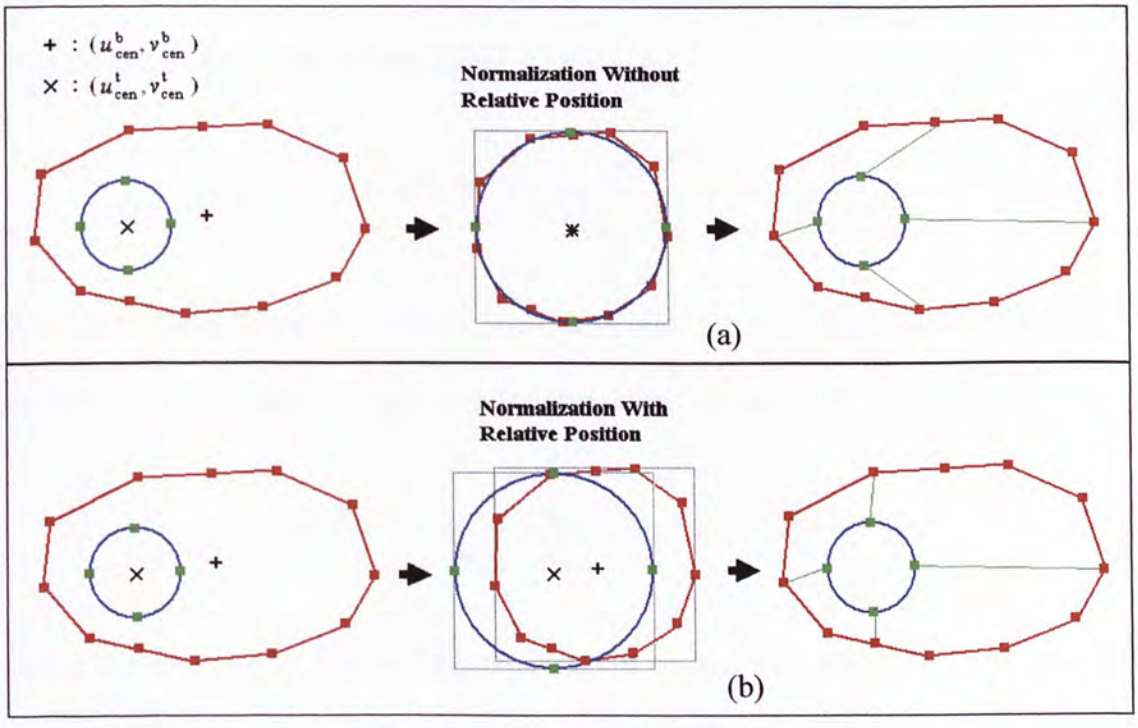

Figure 6.6 Effect of relative position

In order to take into consideration the relative position of the shapes, the distance between the shapes is used for positioning the normalized shapes. The centers of the trim and its bisector loops are respectively expressed as

$$(u^{t}_{cen}, v^{t}_{cen}) = (\frac{u^{t}_{max} - u^{t}_{min}}{2}, \frac{v^{t}_{max} - v^{t}_{min}}{2}) \text{ and}$$

$$(u^{b}_{cen}, v^{b}_{cen}) = (\frac{u^{b}_{max} - u^{b}_{min}}{2}, \frac{v^{b}_{max} - v^{b}_{min}}{2}) \tag{6.3}$$

Denotes the displacement between the shapes' centers as

$$(\Delta u, \Delta v) = (u^{t}_{cen} - u^{b}_{cen}, v^{t}_{cen} - v^{b}_{cen}) \tag{6.4}$$

and assuming the trim shape is inside the bisector loop, the corresponding displacement in the normalized domain is given by

$$(\Delta u'^{b}, \Delta v'^{b}) = (\frac{\Delta u}{u^{b}_{max} - u^{b}_{min}}, \frac{\Delta v}{v^{b}_{max} - v^{b}_{min}}) \tag{6.5}$$

All normalized bisector vertices are translated by the displacement $(\Delta u', \Delta v')$ relative to the normalized trim. After the translation, Equation (6.2) becomes

$$(u'^{b}, v'^{b}) = (\frac{u^{b} - u^{b}_{min}}{u^{b}_{max} - u^{b}_{min}} + \Delta u'^{b}, \frac{v^{b} - v^{b}_{min}}{v^{b}_{max} - v^{b}_{min}} + \Delta v'^{b}) \tag{6.6}$$

Using the example in *Figure 6.6a*, the result of normalizing with relative positioning is shown in *Figure 6.6b*. Obviously, the shapes of the parametric patches are better than those in *Figure 6.6a*.

The above technique can be applied to those trims lying inside the bisector loop. If the bisector loop is inside the trim, i.e., the parametric tile of the boundary trim, their roles are swapped. In this case, Equation (6.5) is rewritten as

$$(\Delta u'^{t}, \Delta v'^{t}) = (\frac{\Delta u}{u^{t}_{max} - u^{t}_{min}}, \frac{\Delta v}{v^{t}_{max} - v^{t}_{min}}) \tag{6.7}$$

And the translation will be applied to the trim. Therefore, Equation (6.1) becomes

$$(u'^{t}, v'^{t}) = (\frac{u^{t} - u^{t}_{min}}{u^{t}_{max} - u^{t}_{min}} + \Delta u'^{t}, \frac{v^{t} - v^{t}_{min}}{v^{t}_{max} - v^{t}_{min}} + \Delta v'^{t}) \quad (6.8)$$

The normalized bisector vertices will be ranked for associating with the most appropriate feature points in the next step.

6.3. Ranking Process

Within a parametric tile, all the valid bisector vertices are identified for every feature point in the previous step. These valid bisector vertices will be ranked so as to determine the appropriate bisector vertex to be associated with the feature point. The ranking is based on two factors. The first one is the distances between a feature point and its valid bisector vertices in the normalized domain. The second one is the interior angles of the bisector vertices in the 3D space. A feature point will be associated with the closest and sharpest bisector vertex. In a parametric tile, given a trim's feature point (u^{t}, v^{t}), the distance from its *i*-th valid bisector vertex (u_i^{b}, v_i^{b}) in the normalized domain is expressed as

$$d_i = \left| (u'^{t}, v'^{t}) - (u_i'^{b}, v_i'^{b}) \right| \quad (6.9)$$

Let the coordinates of the bisector vertex (u_i^{b}, v_i^{b}) in 3D space be $\mathbf{S}(u_i^{b}, v_i^{b})$ and its two adjacent bisector vertices on the same bisector loop be $\mathbf{S}(u_{i0}^{b}, v_{i0}^{b})$ and $\mathbf{S}(u_{i1}^{b}, v_{i1}^{b})$. The interior angle (in 3D space) at the bisector vertex (u_i^{b}, v_i^{b}) can be calculated as

$$\alpha_i = \pi - \cos^{-1}\left(\frac{(\mathbf{S}(u_{i1}^{b}, v_{i1}^{b}) - \mathbf{S}(u_i^{b}, v_i^{b})) \cdot (\mathbf{S}(u_i^{b}, v_i^{b}) - \mathbf{S}(u_{i0}^{b}, v_{i0}^{b}))}{\left|\mathbf{S}(u_{i1}^{b}, v_{i1}^{b}) - \mathbf{S}(u_i^{b}, v_i^{b})\right| \left|\mathbf{S}(u_i^{b}, v_i^{b}) - \mathbf{S}(u_{i0}^{b}, v_{i0}^{b})\right|}\right) \quad (6.10)$$

Combining Equation (6.9) and Equation (6.10), the ranking score of (u_i^b, v_i^b) is defined as

$$s_i = \frac{d_i}{\max(d_j)} + \frac{\alpha_i}{\max(\alpha_j)}, \qquad j = 0, 1, \ldots, n \tag{6.11}$$

where n is the total number of valid bisector vertices being ranked with respect to the feature point (u^t, v^t). Obviously, the lower is the score of the bisector vertex the higher is the ranking of the vertex. The maximum score that a vertex could get is 2.

Recalling from the previous definition, a bisector centroid is a bisector vertex which is shared by three bisectors. Since a bisector centroid is where three parametric tiles meet, the shapes of the tiles at the bisector centroids are irregular in most cases. The interior angles in 3D space at the bisector centroids are usually sharp enough so that the bisector centroids can be considered as the feature points on the bisectors. For this reason, a bisector centroid is given a higher priority in the ranking process. A certain amount of score is deducted from Equation (6.11) if the bisector vertex being ranked is a bisector centroid. This increases the tendency of a feature point to be associated with a bisector centroid. After the process, if there is a bisector centroid which is not found to associate with any feature point, it will be associated with its closest data point on the trim.

6.3.1. Forward and Backward Attachment

The process of associating a trim's feature point to its highest ranked bisector vertex is referred to as "forward attachment". On completion of forward attachment, there may exist bisector centroid not associated with any feature point. The process of associating an un-associated bisector centroid to its closest data point on the trim is

denoted as "backward attachment". The time required for the backward attachment process depends on the number of un-associated bisector centroids on completion of the forward attachment process. Consider a single parametric tile, let the number of feature points on the trim be n^{t} and the number of bisector centroids on the bisector loop be n^{bc}. The number of correspondences built in the forward attachment process should therefore be n^{t}. Let the number of un-associated bisector centroids after the forward attachment be n'^{bc}. Then the number of correspondences needed to be built in the backward attachment process is n'^{bc}. The total number of correspondences built on the tile is thus

$$n^{c} = n^{t} + n'^{bc}, \qquad n'^{bc} \le n^{bc} \tag{6.12}$$

Since the number of parametric patches produced by the parametric tile is proportional to the number of correspondences, it is desired to minimize n^{c} so as to minimize the number of patches. Since n^{t} is fixed, minimizing n^{c} is to minimize n'^{bc}, i.e., to decrease the number of un-associated bisector centroids after the forward attachment process. This is equivalent to maximizing the number of bisector centroids being associated with feature points in the forward attachment process. This is achieved by adjusting the amount to be deducted from Equation (6.11) for ranking bisector centroid.

However, deducting a large amount will cause the feature points to be associated with all bisector centroids ($n'^{bc} = 0$, $n^{c} = n^{t}$). This may result in patches with undesirable shapes. In the proposed system, 5% of the maximum value of s_i is deducted from Equation (6.11). In most cases, this gives patches with regular shapes and the number of patches is satisfactory.

An example is shown in *Figure 6.7*. The data required for establishing correspondence is prepared in all the four parametric tiles. The bisector vertices, including the bisector centroids, are ranked with respect to every feature point of the trims in the forward attachment process. On completion of the forward attachment process, all the feature points are associated with the bisector vertices with the highest ranking. Some bisector centroids are left un-associated, as shown in *Figure 6.8*. The backward attachment process associates these bisector centroids with their nearest data points on the trims (*Figure 6.9*).

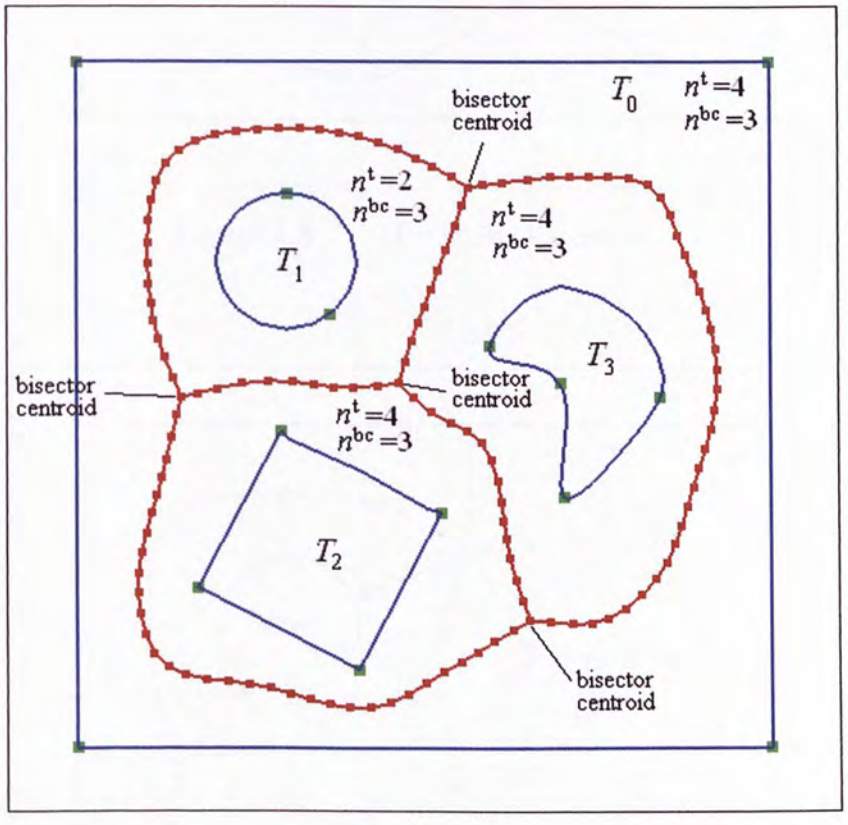

Figure 6.7 Data for correspondence establishment

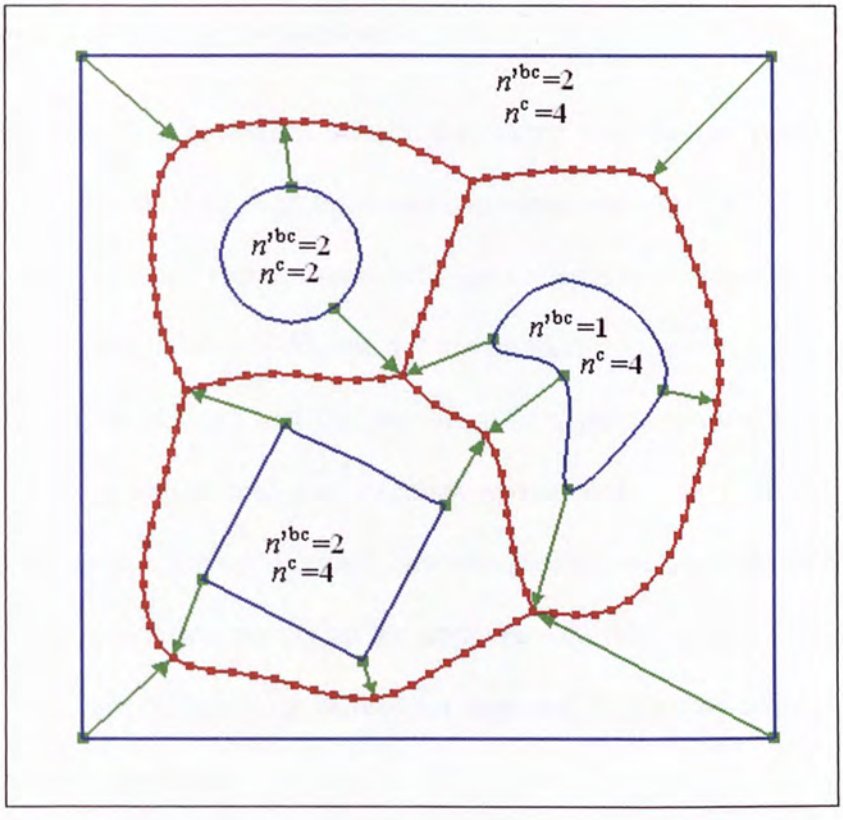

Figure 6.8 Forward attachment

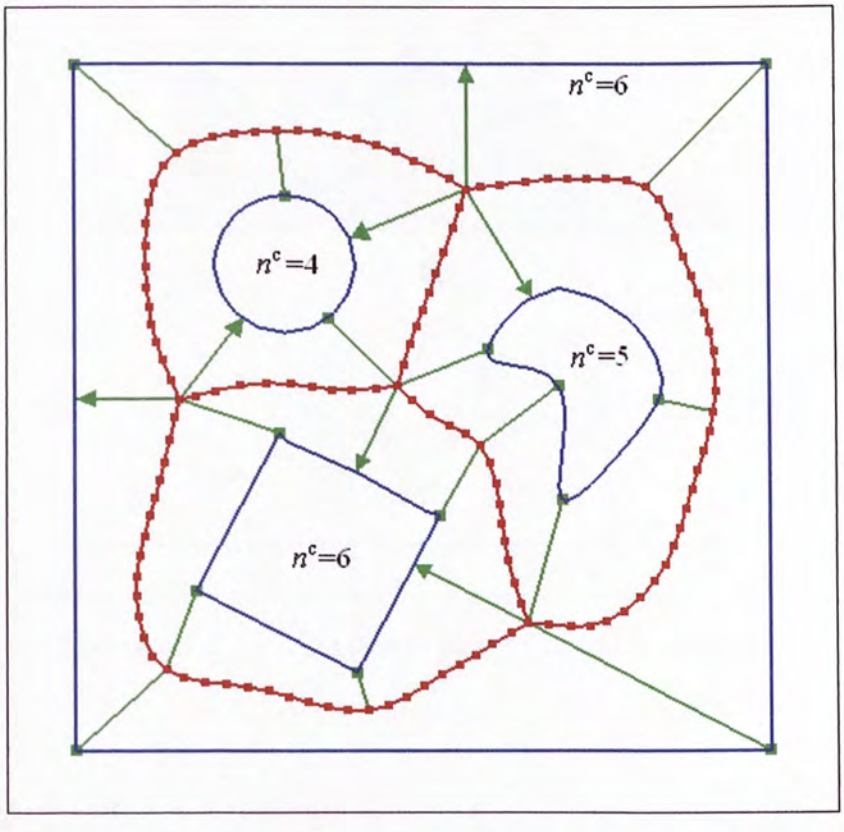

Figure 6.9 Backward attachment

6.3.2. Singly Linked Bisector Vertices

After the forward and backward attachment, every trim feature point or bisector centroid should be attached to at least one correspondence link. Continuity problem may occur at the bisector vertices with only one correspondence link. In *Figure 6.10*, some of the parametric tiles in *Figure 6.9* are divided into parametric patches along the links. It can be noticed that the boundary of a patch is formed by a bisector segment, a trim segment and two correspondence links. At those singly linked bisector vertices, the tiles are divided into two patches on one side of the bisector segment, while no division occurs on the opposite side. This means that the adjacent patches do not share the same set of bisector segments as their boundaries, leading to possible continuity problem.

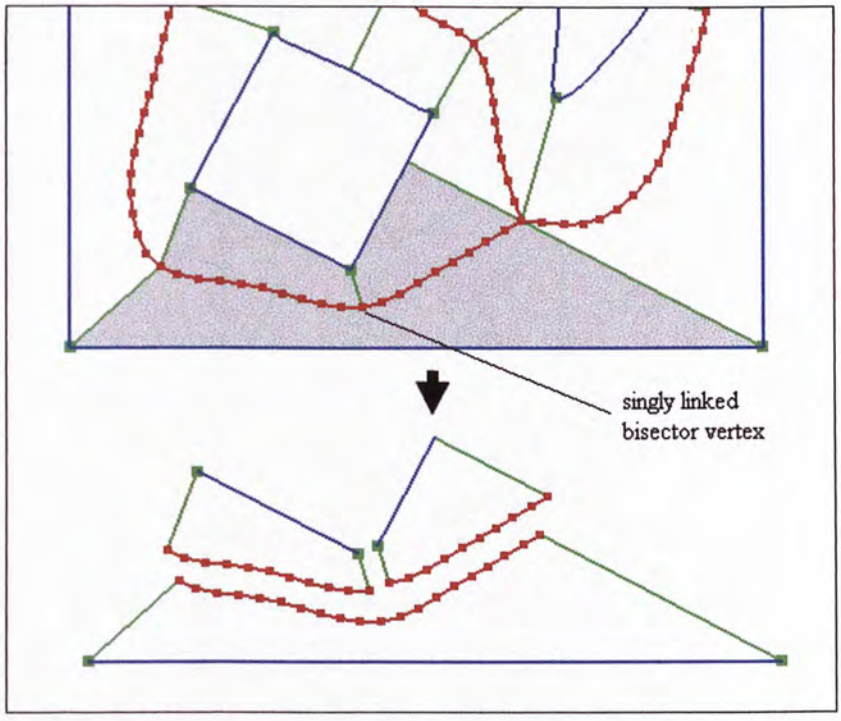

Figure 6.10 Patches at singly linked bisector vertex

To solve this problem, the singly linked bisector vertices are treated as bisector centroids in the process of the correspondence establishment. In the forward attachment process, these vertices will be given higher priority, and hence higher tendency for associating with the trims feature points. Similarly, in the backward attachment process, any singly linked bisector vertices will be associated with the nearest trims data point so that each of them is finally attached to two correspondence links on both sides of the bisector loops. The result of the example in *Figure 6.7* to *Figure 6.9* is shown in *Figure 6.11* in which all the divided parametric patches share with their adjacent patches the same set of bisector segments as their boundaries.

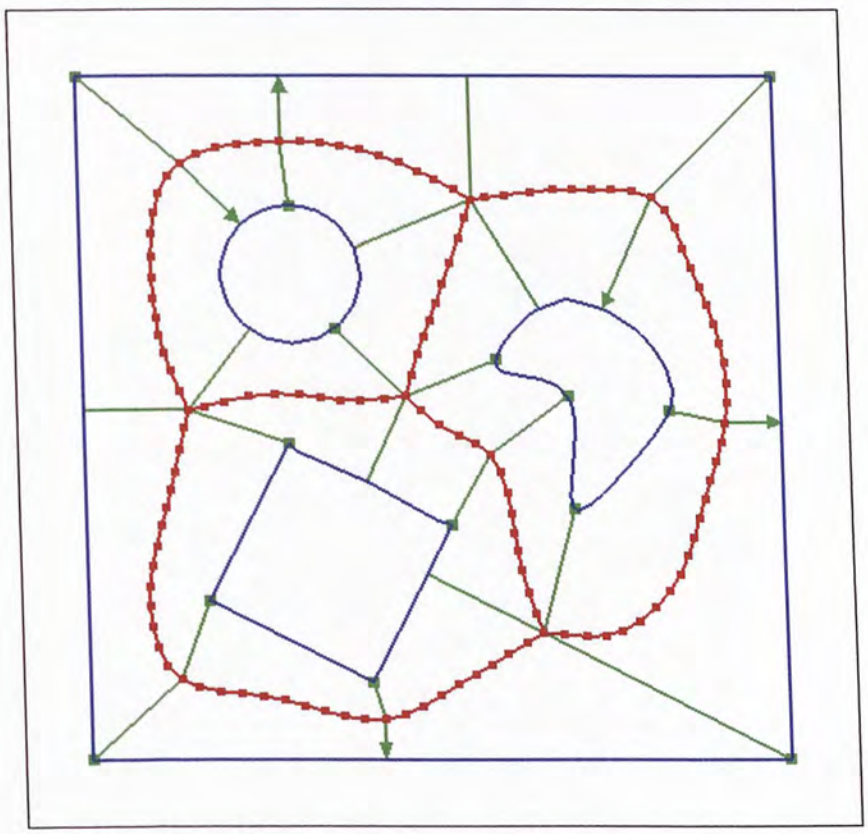

Figure 6.11 Resolving singly linked bisector vertices

Throughout the process of correspondence establishment, tests are performed to avoid cross correspondence, i.e., a link intersecting with other links. In case there is no feature point and bisector centroid in a parametric tile, any two bisector vertices, which are half the bisector perimeter apart, will be linked to the nearest data points of the trim.

Chapter 7. Surface Fitting

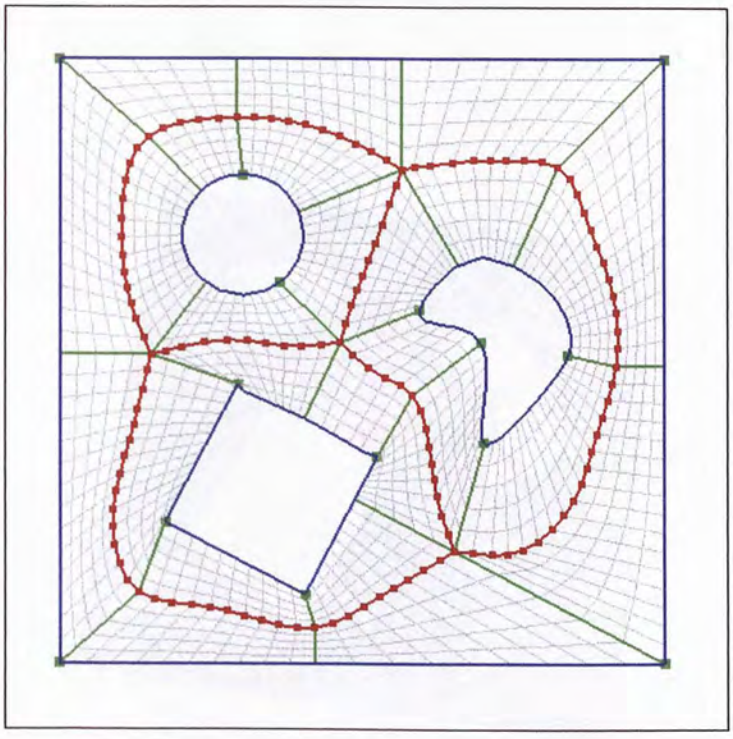

Figure 7.1 Grids in all parametric patches

Up to this stage, feature points on the trims and their corresponding bisector vertices are extracted. A pair of consecutive feature points defines a segment of the trimming curve. The bisector vertices corresponding to these feature points define a corresponding bisector segment. A trim segment and a bisector segment, together with a pair of correspondence links connecting the feature points and their corresponding bisector vertices define a parametric region of a regular surface. This parametric region is referred to as a parametric patch. In every patch region, parametric grid is constructed (*Figure 7.1*). Surface points are generated at the grid points for the subsequent surface fitting. Points on the surface corresponding to the grid points can be obtained by substituting the grid points' parametric coordinates into Equation (3.1). Using chord-length parameterization, a surface defined by each

parametric region is then constructed (*Figure 7.2*). Apart from the process of surface fitting, the factors affecting the continuity between adjacent patches will also be discussed in the following sections.

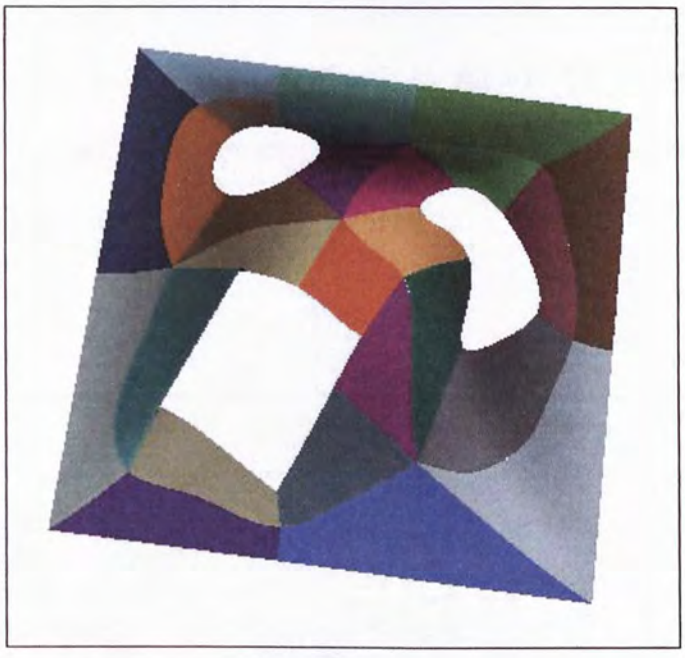

Figure 7.2 Surface patches in three-dimensional space

7.1. Parametric Patches

Given a parametric tile with correspondence established between the shapes of the trim and the bisector loop, it can be broken down into a set of parametric patches by dividing the tile along the correspondence link. Each of these parametric patches represents a specific region on the valid region of the original trimmed surface. The union of the surface patches represented by the parametric patches approximates the trimmed surface. Therefore, the first step in the surface fitting process is to define the parametric patches inside every parametric tile.

7.1.1. Parametric Tile and Patch

After the establishment of correspondence between the bisector vertices and the trim's feature points, there are correspondence links which run across each parametric tile. Each parametric tile is divided into a set of parametric patches by cutting along the correspondence links. In the example shown in *Figure 7.3*, the parametric tile is broken down into a set of parametric patches. There are six correspondence links within the tile and thus six patches are developed.

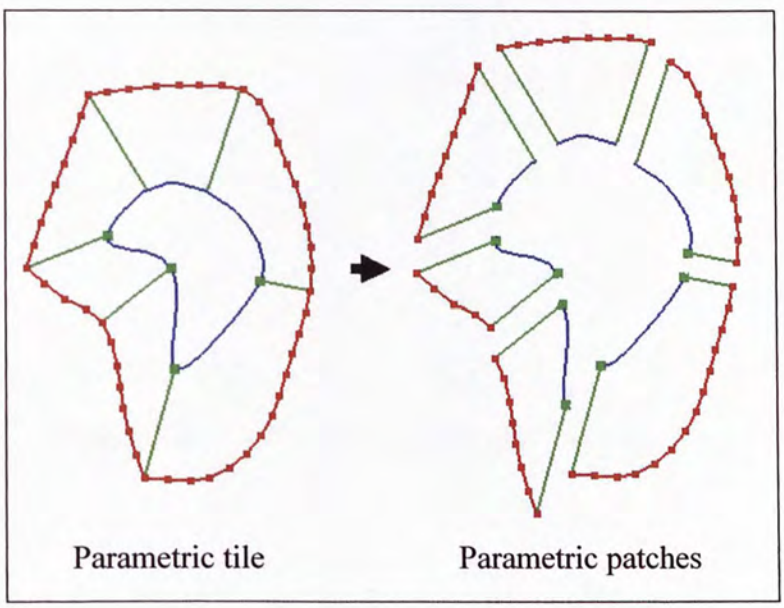

Figure 7.3 Division of parametric tile

The boundary of a parametric patch is composed of four parts: a trim segment, a bisector segment and two adjacent correspondence links. The trim segments are constructed by dividing the trim at the points of correspondence. The bisector segments are constructed by dividing the bisector loop at the points of correspondence.

7.1.2. Local Parametric Coordinate System

After every parametric tile is broken down into a set of parametric patches, each of the patches will be processed individually. First, a local parametric coordinate system is assigned to every patch so as to facilitate the parametric grid generation in the later step. The *u*-coordinate of the patch is aligned along the bisector segment and the trim segment. The *v*-coordinate is aligned along the two correspondence links. *Figure 7.4* shows the local parametric coordinate system of the parametric patch.

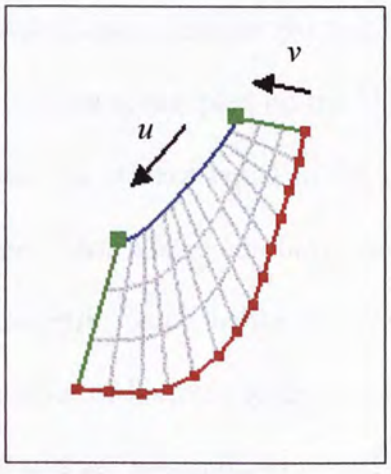

Figure 7.4 Local parametric coordinate system

7.2. Parametric Grids

Parametric grid is generated within every parametric patch according to the local coordinate system defined in the previous section. To generate the grid, the following information is necessary to be specified or determined:

n: the number of sample points along the *u*-direction

p: the order in the *u*-direction

m: the number of sample points along the *v*-direction

q: the order in the *v*-direction

Interpolating a B-Spline surface through the $n \times m$ sample points requires the following conditions to be satisfied

$$n \geq p, \quad m \geq q \tag{7.1}$$

7.2.1. Sample Points on the Patch Boundary

The number of the sample points in the u-direction can be simply set equal to the number of the bisector vertices on the bisector segment. If the number of bisector vertices on the bisector segment is not sufficient for satisfying condition (7.1), evenly distributed bisector vertices will be re-sampled on the bisector segment such that the number of the bisector vertices is at least equal to the order in the u-direction. An example is shown in *Figure 7.5a*. There are only three bisector vertices on the bisector segment of the parametric patch in the figure. If the specified u-order is greater than 3, e.g. 6, the number of bisector vertices is not enough. In this case, six bisector vertices are required and are re-sampled on the bisector segment as shown in *Figure 7.5b*. For the neighbouring patch which shares the same original bisector segment as boundary, its bisector segment will be replaced by the new bisector segment (*Figure 7.6*).

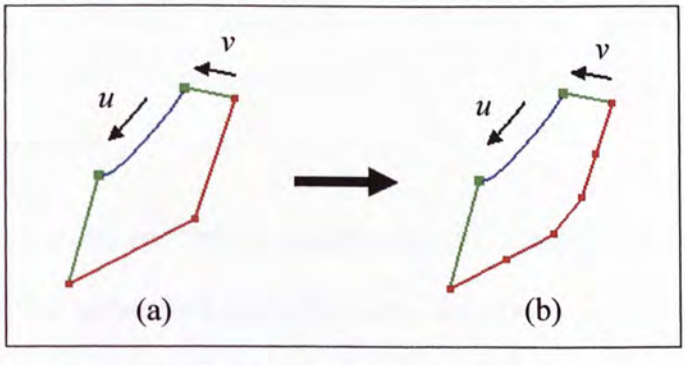

Figure 7.5 Re-sampling of bisector vertices

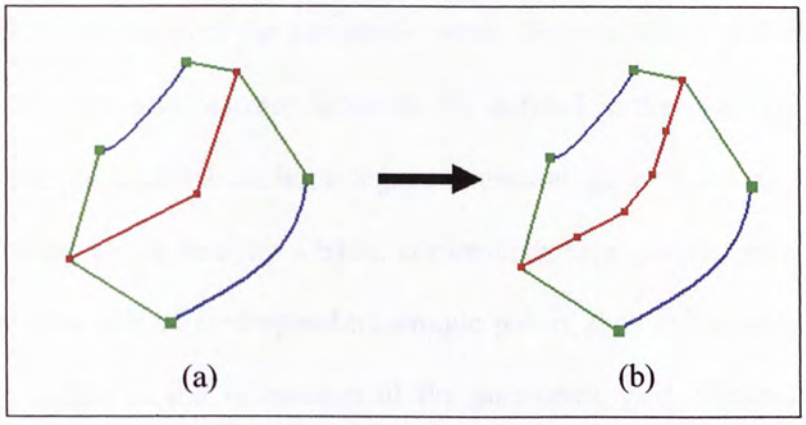

Figure 7.6 Neighbouring patches with re-sampled bisector vertices

Given the set of bisector vertices on the bisector segment, the same number of data points will be uniformly sampled on the trim segment (*Figure 7.7*). Each sample point on the trim segment corresponds to a bisector vertex on the bisector segment.

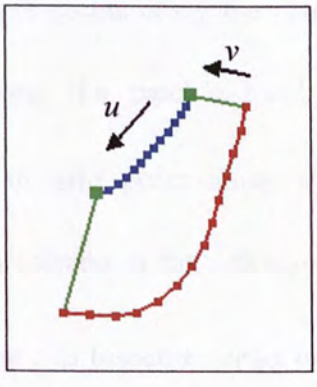

Figure 7.7 Sample points on trim segment

7.2.2. Grid Generation

As discussed in previous sections, a parametric grid is generated in each parametric patch. A point of the parametric grid represents the parametric coordinates of a data point on the surface patch in the 3D space.

Along the u-direction of the parametric patch, data points are uniformly sampled on the trim segment and bisector segment. As defined in the previous section, the numbers of sample points n on both segments are the same. A sample point on the bisector segment, i.e., a bisector vertex, corresponds to a sample point on the trim segment. For each pair of corresponding sample points, they are linearly interpolated to obtain the points in the v-direction of the parametric grid. Similarly, m sample points are generated through linear interpolation along the v-direction.

To compute the parametric coordinates of the grid points,

$$(u_{i,j}^{g}, v_{i,j}^{g}) = (u_i^{t}, v_i^{t})t_j + (u_i^{b}, v_i^{b})(1-t_j),$$

$$i = 0,1,...,n-1, \quad j = 0,1,...,m-1, \quad t_j \in [0,1] \tag{7.2}$$

where n is the number of sample points along the patch's local u-direction, m is the number of sample points along the patch's local v-direction, $(u_{i,j}^{g}, v_{i,j}^{g})$ is the coordinate of the i-th and j-th grid point along the patch's u- and v-direction respectively, (u_i^{t}, v_i^{t}) is the coordinate of the i-th sample point on the trim segment, (u_i^{b}, v_i^{b}) is the coordinate of the i-th bisector vertex on the bisector segment, and t_j is the j-th parameter value for the linear interpolation between (u_i^{t}, v_i^{t}) and (u_i^{b}, v_i^{b}). All the parametric coordinates of (u_i^{t}, v_i^{t}), (u_i^{b}, v_i^{b}) and $(u_{i,j}^{g}, v_{i,j}^{g})$ are specified relative to the parametric coordinate system of the original trimmed surface. The total number of grid points in the grids is $n \times m$. Because of the uniform distribution of data points along the v-direction, t_j can be easily calculated by

$$t_j = j(\frac{1}{m-1}), \qquad j = 0, 1, ..., m-1 \tag{7.3}$$

Figure 7.8 shows a result of grid generation by linear interpolation. It is an example with $n = 14$ and $m = 4$, and a total of 56 grid points in the patch.

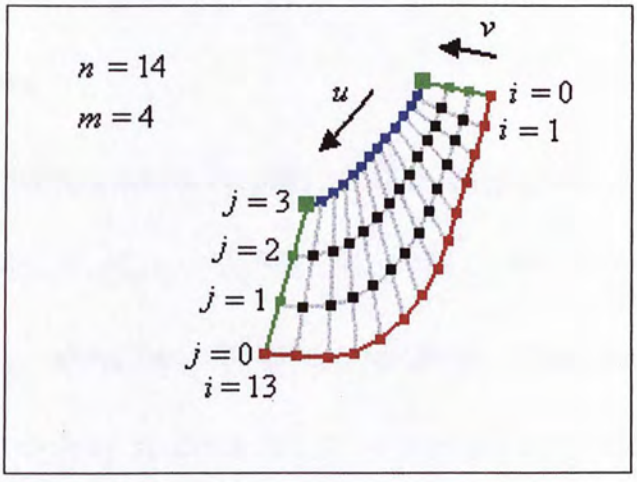

Figure 7.8 Grid points by linear interpolation

Since the coordinates of the grid points $(u^g_{i,j}, v^g_{i,j})$ are expressed relative to the coordinate system of the original parametric trimmed surface, the 3D coordinates of the grid points on the surface patch can be obtained by substituting the parametric coordinates into the representation of the 3D trimmed surface in Equation (3.1). The grid point in 3D space is thus given by

$$S(u^g_{i,j}, v^g_{i,j}) = (x(u^g_{i,j}, v^g_{i,j}), y(u^g_{i,j}, v^g_{i,j}), z(u^g_{i,j}, v^g_{i,j})) = (x^g_{i,j}, y^g_{i,j}, z^g_{i,j})$$

$$i = 0, 1, ..., n-1, \quad j = 0, 1, ..., m-1, \quad t_j \in [0, 1] \tag{7.4}$$

These $n \times m$ 3D grid points will then be used as the data points for the surface fitting process.

7.3 Surface Patches Construction

In the final step of the surface fitting process, control points of the surface patch defined by each parametric patch region are computed. In this thesis, the surface patches are represented by B-Spline surfaces.

7.3.1. Knot Vectors

Consider the data points given by Equation (7.4). Assuming a fixed j, the n data points $(x^{g}_{0,j}, y^{g}_{0,j}, z^{g}_{0,j})$, $(x^{g}_{1,j}, y^{g}_{1,j}, z^{g}_{1,j})$, ..., $(x^{g}_{n-1,j}, y^{g}_{n-1,j}, z^{g}_{n-1,j})$ will have parametric values $\bar{u}_{0,j}$, $\bar{u}_{1,j}$, ..., $\bar{u}_{n-1,j}$ along the u-direction respectively. These parametric values are calculated by the method of chord length parameterization [10]. The total chord length between the data points is denoted as

$$l_j = \sum_{i=1}^{n-1} \left| (x^{g}_{i,j}, y^{g}_{i,j}, z^{g}_{i,j}) - (x^{g}_{i-1,j}, y^{g}_{i-1,j}, z^{g}_{i-1,j}) \right|,$$

$$j = 0, 1, ..., m-1 \qquad (7.5)$$

Setting $\bar{u}_{0,j} = 0$ and $\bar{u}_{n-1,j} = 1$, the parametric values in the other u-intervals can be obtained by

$$\bar{u}_{i,j} = \bar{u}_{i-1,j} + \frac{\left| (x^{g}_{i,j}, y^{g}_{i,j}, z^{g}_{i,j}) - (x^{g}_{i-1,j}, y^{g}_{i-1,j}, z^{g}_{i-1,j}) \right|}{l_j},$$

$$i = 1, 2, ..., n-2, \quad j = 0, 1, ..., m-1 \qquad (7.6)$$

For every fixed i, the parametric values $\bar{u}_i$ are computed by the expression

$$\bar{u}_i = \frac{1}{m} \sum_{j=0}^{m-1} \bar{u}_{i,j}, \qquad i = 0, 1, ..., n-1 \qquad (7.7)$$

The knots of the knot vector in the u-direction are specified as

$$u_0 = u_1 = \ldots = u_{p-1} = 0,$$

$$u_n = u_{n+1} = \ldots = u_{n+p-1} = 1,$$

$$u_{k+p-1} = \frac{1}{p-1} \sum_{i=k}^{k+p-2} \bar{u}_i, \qquad k = 1, 2, \ldots, n-p \tag{7.8}$$

and the length of the knot vector is $n+p$. The knot vector in the v-direction is computed with a similar approach.

$$v_0 = v_1 = \ldots = v_{q-1} = 0,$$

$$v_m = v_{m+1} = \ldots = v_{m+q-1} = 1,$$

$$v_{k+q-1} = \frac{1}{q-1} \sum_{j=k}^{k+q-2} \bar{v}_j, \qquad k = 1, 2, \ldots, m-q \tag{7.9}$$

where

$$\bar{v}_j = \frac{1}{n} \sum_{i=0}^{n-1} \bar{v}_{i,j}, \qquad j = 0, 1, \ldots, m-1 \tag{7.10}$$

7.3.2. Control Vertices

With the knot vectors along the u and v-directions of the patch, the data points and their parametric values with respect to the knot vectors, the control vertices of a B-Spline surface interpolating the data points are expressed as

$$\mathbf{D} = \mathbf{N}_u \cdot \mathbf{P} \cdot \mathbf{N}_v \tag{7.11}$$

where

$$\mathbf{D} = \begin{bmatrix} (x^{g}_{0,0}, y^{g}_{0,0}, z^{g}_{0,0}) & (x^{g}_{0,1}, y^{g}_{0,1}, z^{g}_{0,1}) & \cdots & (x^{g}_{0,m-1}, y^{g}_{0,m-1}, z^{g}_{0,m-1}) \\ (x^{g}_{1,0}, y^{g}_{1,0}, z^{g}_{1,0}) & (x^{g}_{1,1}, y^{g}_{1,1}, z^{g}_{1,1}) & \cdots & (x^{g}_{1,m-1}, y^{g}_{1,m-1}, z^{g}_{1,m-1}) \\ \vdots & \vdots & \ddots & \vdots \\ (x^{g}_{n-1,0}, y^{g}_{n-1,0}, z^{g}_{n-1,0}) & (x^{g}_{n-1,1}, y^{g}_{n-1,1}, z^{g}_{n-1,1}) & \cdots & (x^{g}_{n-1,m-1}, y^{g}_{n-1,m-1}, z^{g}_{n-1,m-1}) \end{bmatrix} \tag{7.12}$$

is the $n \times m$ matrix of the grid points on the surface patch,

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_{0,0} & \mathbf{P}_{0,1} & \cdots & \mathbf{P}_{0,m-1} \\ \mathbf{P}_{1,0} & \mathbf{P}_{1,1} & \cdots & \mathbf{P}_{1,m-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{n-1,0} & \mathbf{P}_{n-1,1} & \cdots & \mathbf{P}_{n-1,m-1} \end{bmatrix} \tag{7.13}$$

is the $n \times m$ matrix of the surface patch's control vertices,

$$\mathbf{N}_u = \begin{bmatrix} N_{0,p}(\overline{u}_0) & N_{1,p}(\overline{u}_0) & \cdots & N_{n-,p}(\overline{u}_0) \\ N_{0,p}(\overline{u}_1) & N_{1,p}(\overline{u}_1) & \cdots & N_{n-1,p}(\overline{u}_1) \\ \vdots & \vdots & \ddots & \vdots \\ N_{0,p}(\overline{u}_{n-1}) & N_{1,p}(\overline{u}_{n-1}) & \cdots & N_{n-1,p}(\overline{u}_{n-1}) \end{bmatrix} \tag{7.14}$$

is the $n \times n$ matrix of the basis functions in the *u*-direction of the surface patch, and

$$\mathbf{N}_v = \begin{bmatrix} N_{0,q}(\overline{v}_0) & N_{0,q}(\overline{v}_1) & \cdots & N_{0,q}(\overline{v}_{m-1}) \\ N_{1,q}(\overline{v}_0) & N_{1,q}(\overline{v}_1) & \cdots & N_{1,q}(\overline{v}_{m-1}) \\ \vdots & \vdots & \ddots & \vdots \\ N_{m-1,q}(\overline{v}_0) & N_{m-1,q}(\overline{v}_1) & \cdots & N_{m-1,q}(\overline{v}_{m-1}) \end{bmatrix} \tag{7.15}$$

is the $m \times m$ matrix of the basis functions in the *v*-direction of the surface patch. By multiplying the inverse of the basis function matrices on both side of Equation (7.12), the unknown control vertices of the surface patch can be solved.

$$\begin{aligned} \mathbf{N}_u \cdot \mathbf{P} \cdot \mathbf{N}_v &= \mathbf{D} \\ \mathbf{N}_u^{-1} \cdot \mathbf{N}_u \cdot \mathbf{P} \cdot \mathbf{N}_v \cdot \mathbf{N}_v^{-1} &= \mathbf{N}_u^{-1} \cdot \mathbf{D} \cdot \mathbf{N}_v^{-1} \\ \mathbf{P} &= \mathbf{N}_u^{-1} \cdot \mathbf{D} \cdot \mathbf{N}_v^{-1} \end{aligned} \tag{7.16}$$

Figure 7.9 shows a trimmed surface in the parametric space. In the figure, parametric grids are generated for all parametric patches. For the highlighted patch in the figure, the corresponding 3D surface patch is shown in *Figure 7.10*. The grey wire-frame in the figure is the wire-frame of the original trimmed surface. The brown vertices and straight lines are respectively the control vertices and the control mesh of the surface patch. The same process is performed on all the other parametric patches of the trimmed surface to determine the corresponding surface patches defined. The union of all the 3D surface patches approximates the original trimmed surface (*Figure 7.2*).

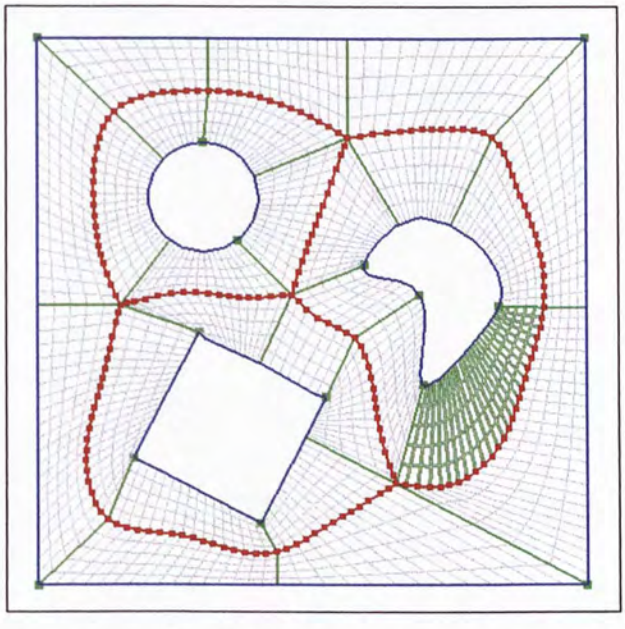

Figure 7.9 Patch in parametric space (highlighted)

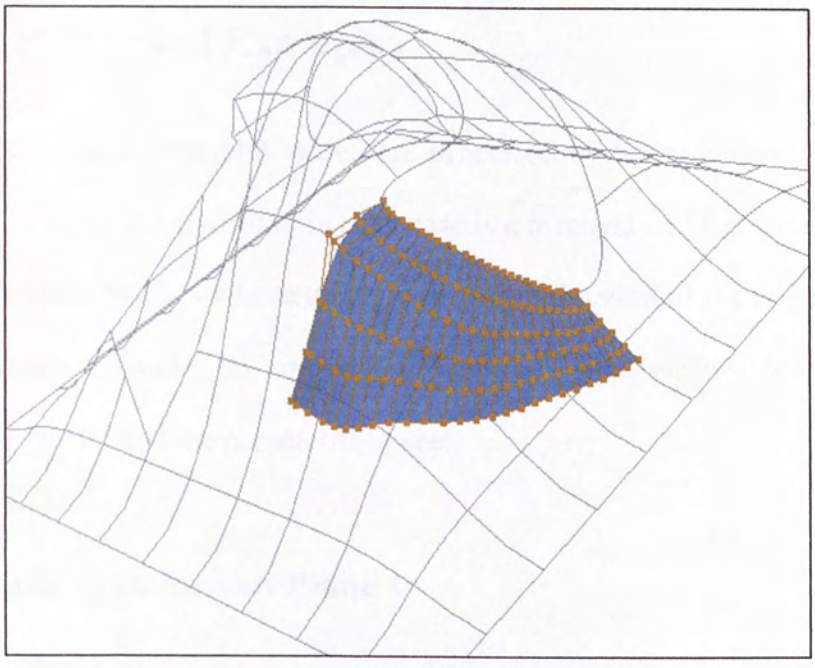

Figure 7.10 Surface patch in 3D space

Chapter 8. Worked Examples

In this chapter, some examples which are processed by the proposed algorithm are shown. The input to the algorithm in each case is a trimmed surfaces and the output is a group of surface patches whose union is the approximation of the original trimmed surface. In each example, the original surface and output patches of the model is illustrated in the 3D and the parametric space.

8.1. Example 1: Deformed Plane 1

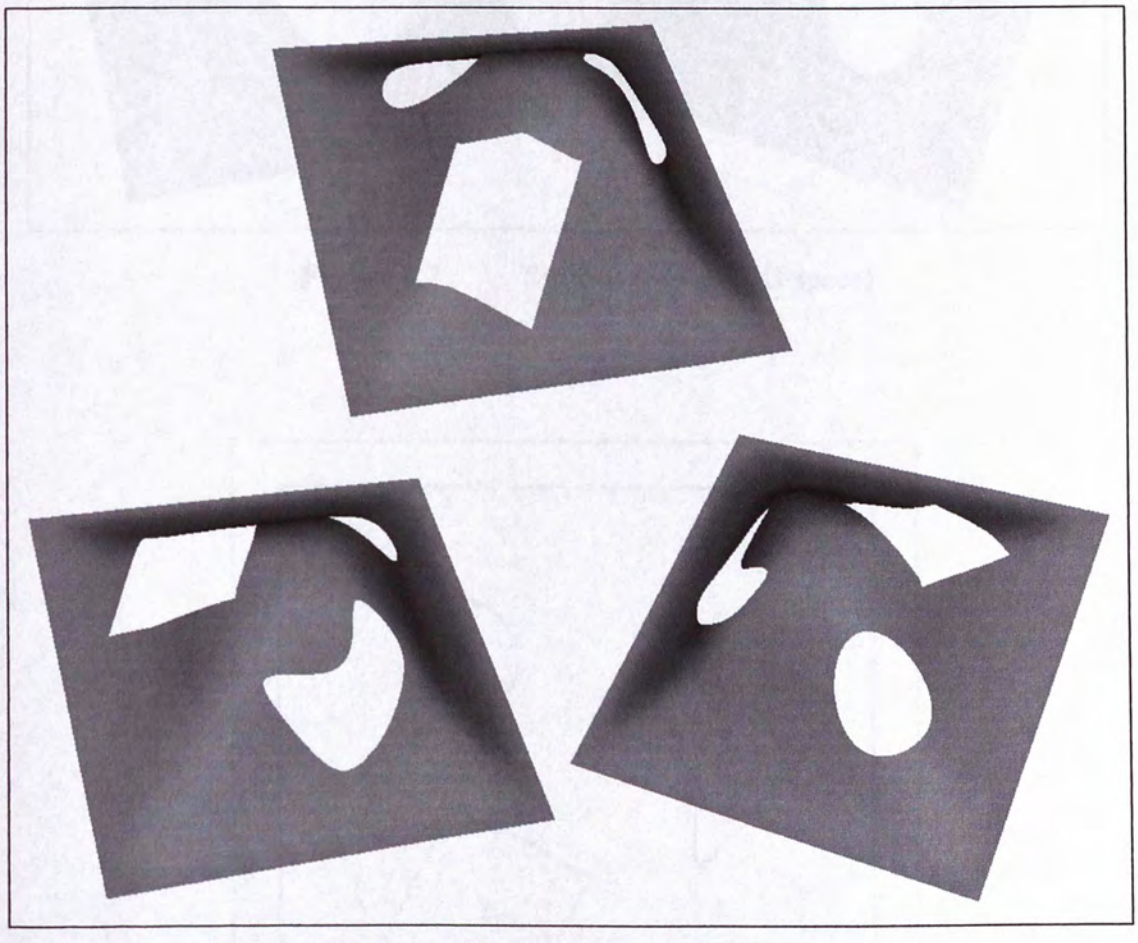

Figure 8.1 Original trimmed surface (3D space)

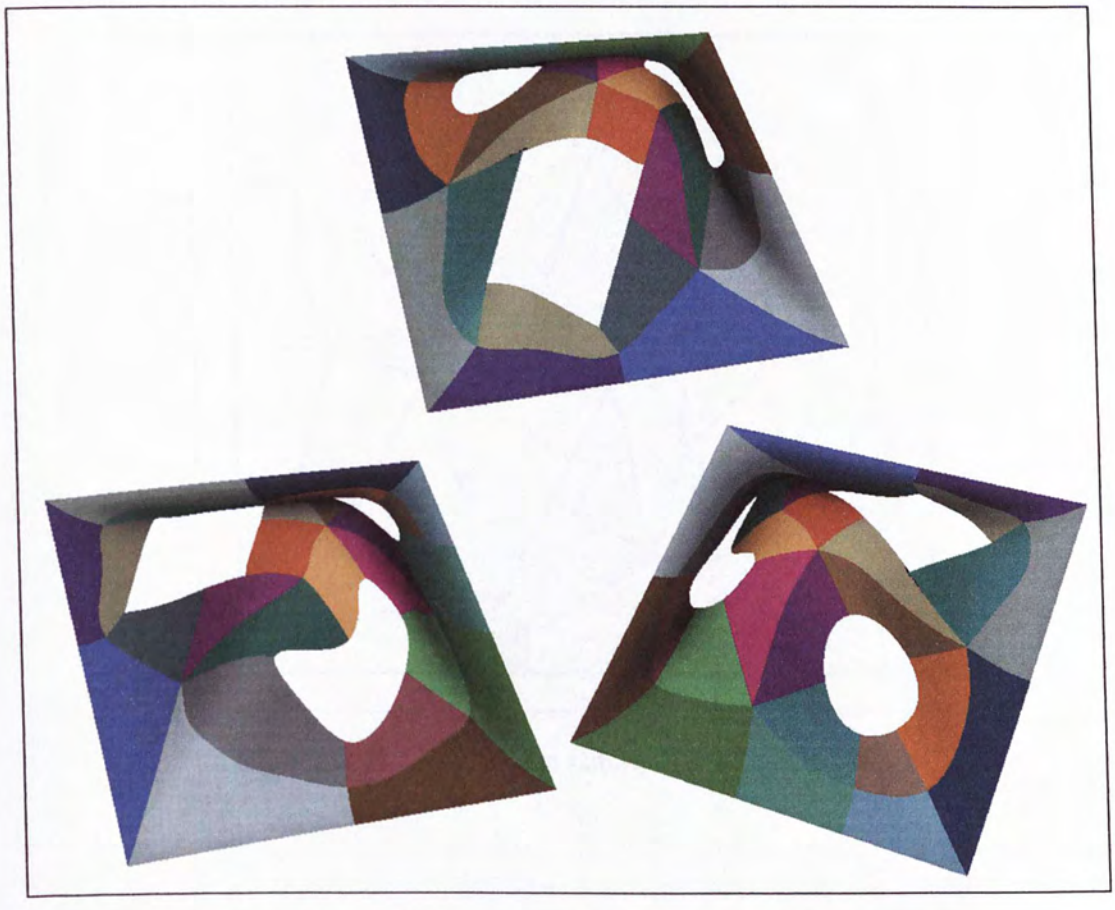

Figure 8.2 Surface patches (3D space)

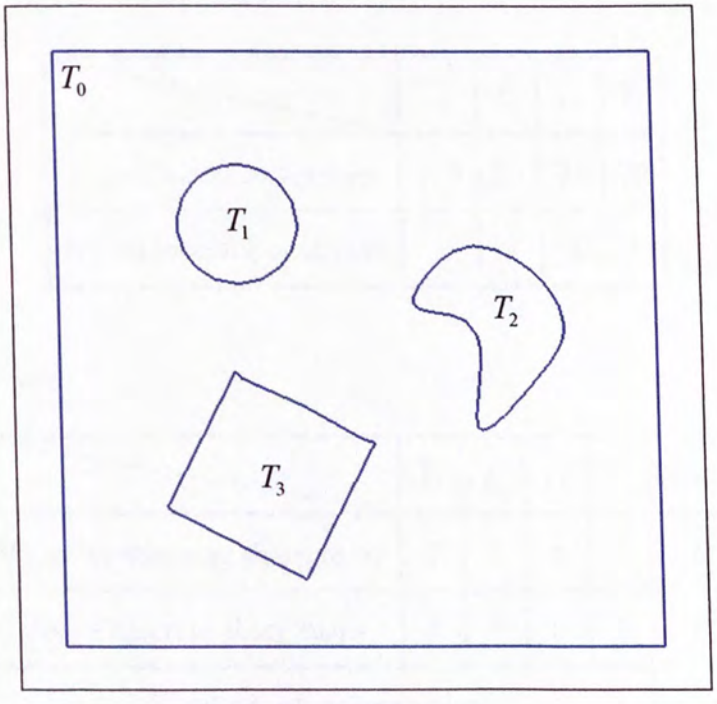

Figure 8.3 Original trimmed surface (parametric space)

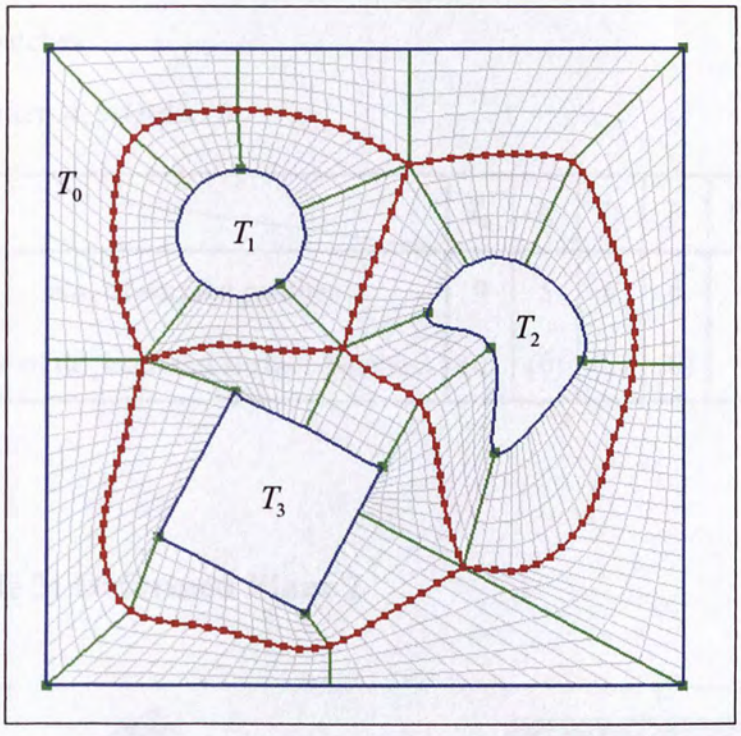

Figure 8.4 Surface patches (parametric space)

Details of Example 1:

1. Voronoï diagram

   - Initial degree of triangulation: 8

	T_0	T_1	T_2	T_3
No. of bisector vertices	118	59	73	70
No. of bisector centroids	3	3	3	3

2. Feature points

	T_0	T_1	T_2	T_3	Total
No. of continuous sharp turns	0	2	4	0	6
No. of discrete sharp turns	4	0	0	4	8
Total no. of feature points					14

3. Surface patches

- u-Order: 4, v-Order: 6

	T_0	T_1	T_2	T_3	Total
No. of surface patches	9	5	6	6	26
(No. of degenerated surface patches)	(0)	(0)	(0)	(0)	(0)

8.2. Example 2: Deformed Plane 2

Figure 8.5 Original trimmed surface (3D space)

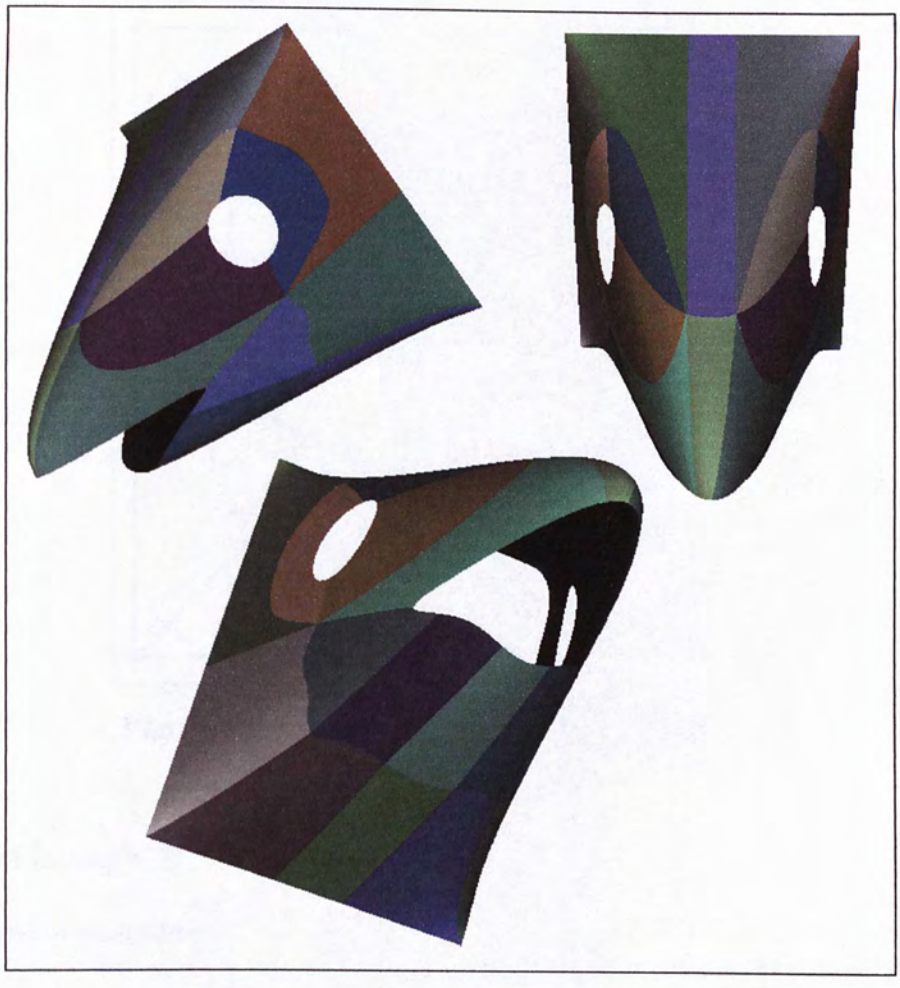

Figure 8.6 Surface patches (3D space)

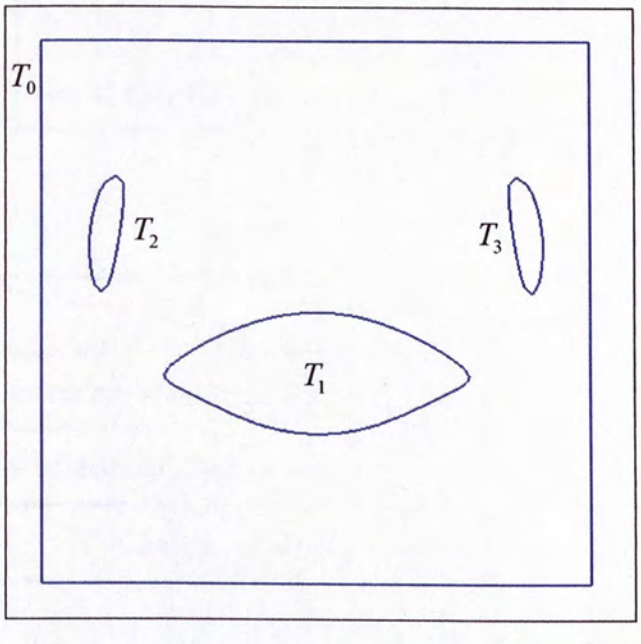

Figure 8.7 Original trimmed surface (parametric space)

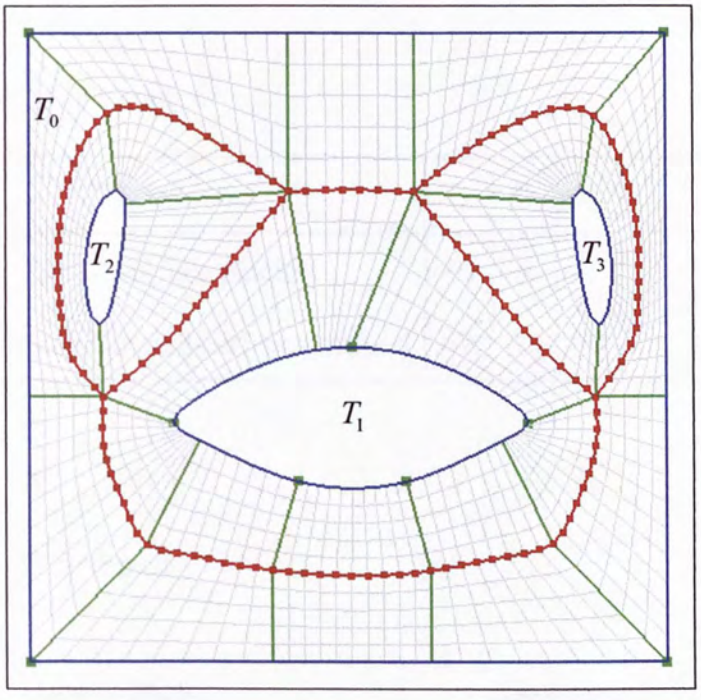

Figure 8.8 Surface patches (parametric space)

Details of Example 2:

1. Voronoï diagram

   - Initial degree of triangulation: 8

	T_0	T_1	T_2	T_3
No. of bisector vertices	116	87	49	48
No. of bisector centroids	4	4	2	2

2. Feature points

	T_0	T_1	T_2	T_3	Total
No. of continuous sharp turns	0	5	0	0	5
No. of discrete sharp turns	4	0	0	0	4
Total no. of feature points					9

3. Surface patches

- u-Order: 4, v-Order: 6

	T_0	T_1	T_2	T_3	Total
No. of surface patches	10	8	3	3	24
(No. of degenerated surface patches)	(0)	(0)	(0)	(0)	(0)

8.3. Example 3: Sphere

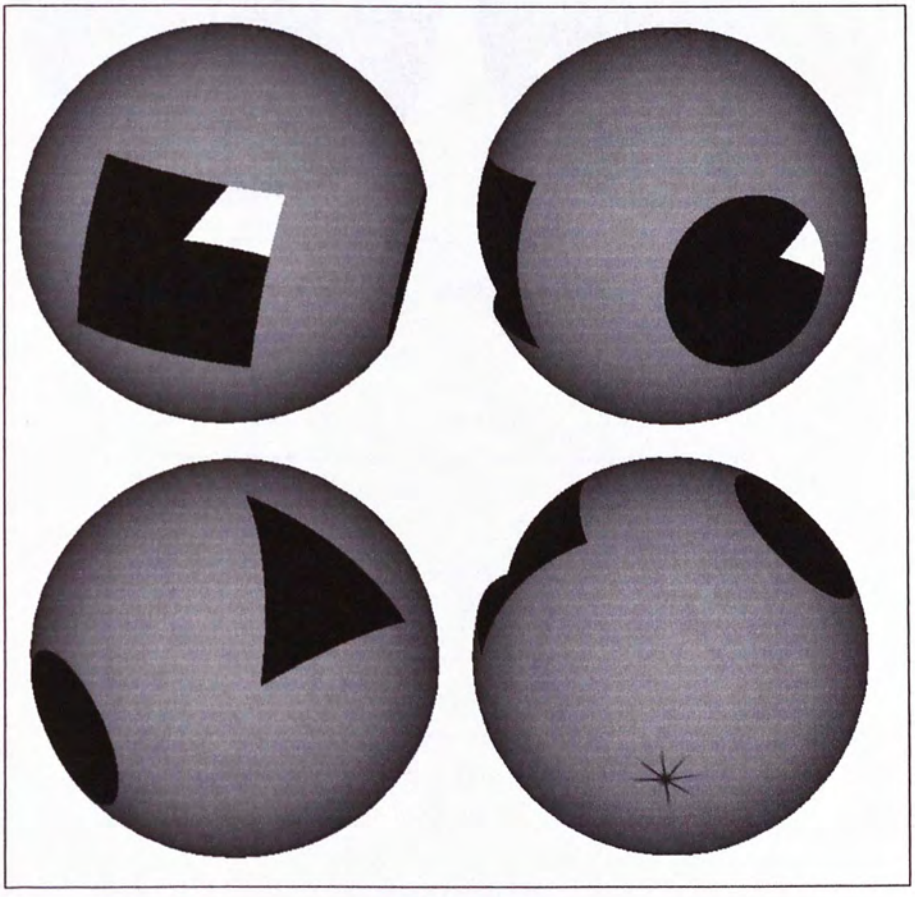

Figure 8.9 Original trimmed surface (3D space)

Figure 8.10 Surface patches (3D space)

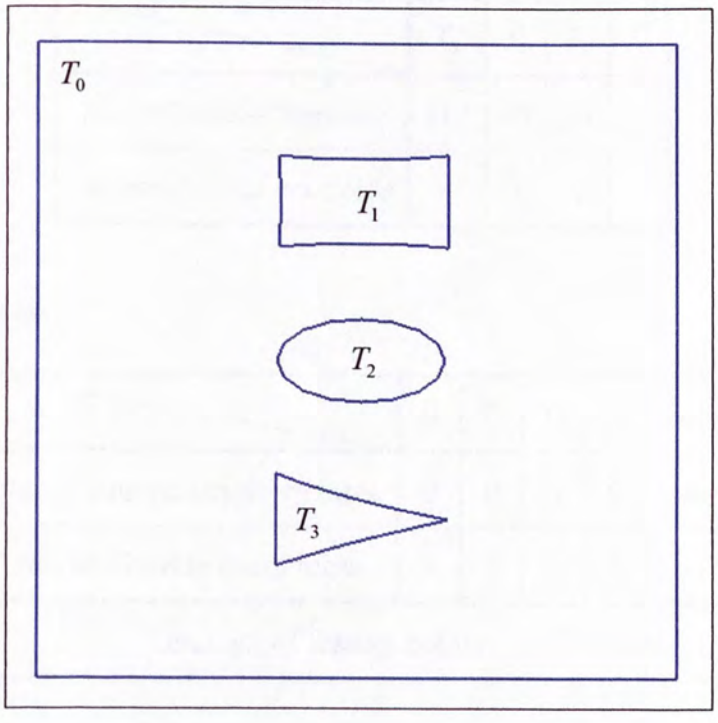

Figure 8.11 Original trimmed surface (parametric space)

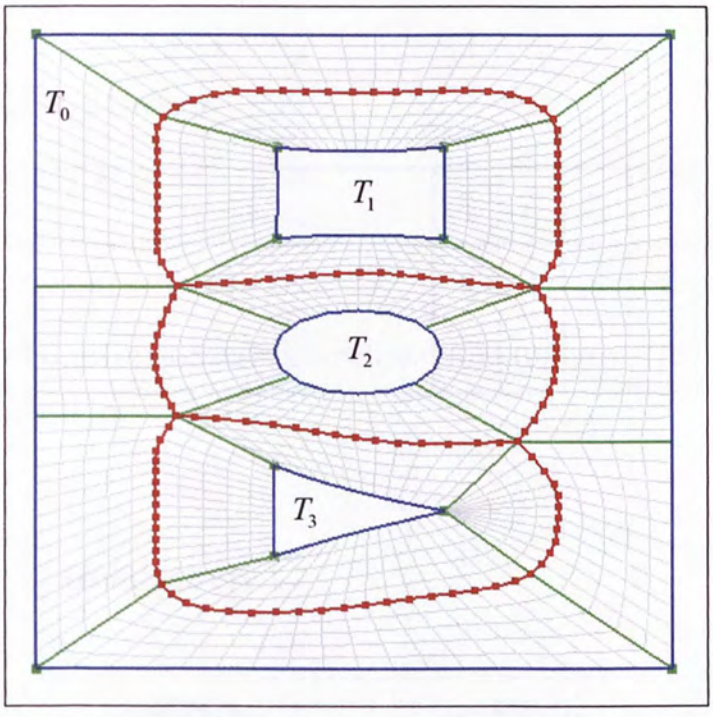

Figure 8.12 Surface patches (parametric space)

Details of Example 3:

1. Voronoï diagram

   - Initial degree of triangulation: 8

	T_0	T_1	T_2	T_3
No. of bisector vertices	117	65	59	55
No. of bisector centroids	4	2	4	2

2. Feature points

	T_0	T_1	T_2	T_3	Total
No. of continuous sharp turns	0	0	0	0	0
No. of discrete sharp turns	4	4	0	3	11
Total no. of feature points					11

3. Surface patches

- u-Order: 4, v-Order: 6

	T_0	T_1	T_2	T_3	Total
No. of surface patches	8	4	4	4	20
(No. of degenerated surface patches)	(0)	(0)	(0)	(1)	(1)

8.4. Example 4: Hemisphere 1

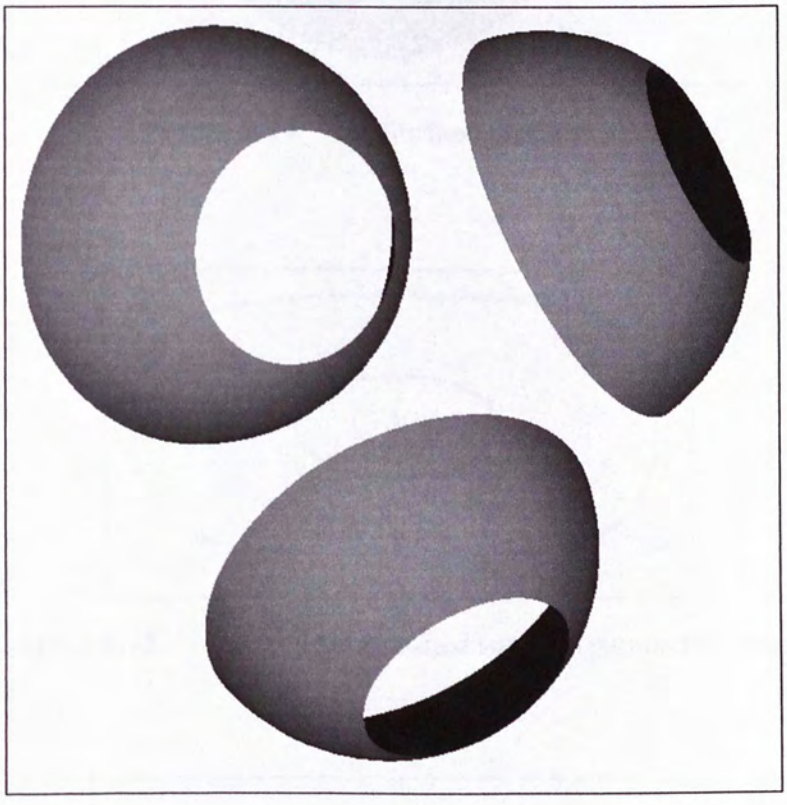

Figure 8.13 Original trimmed surface (3D space)

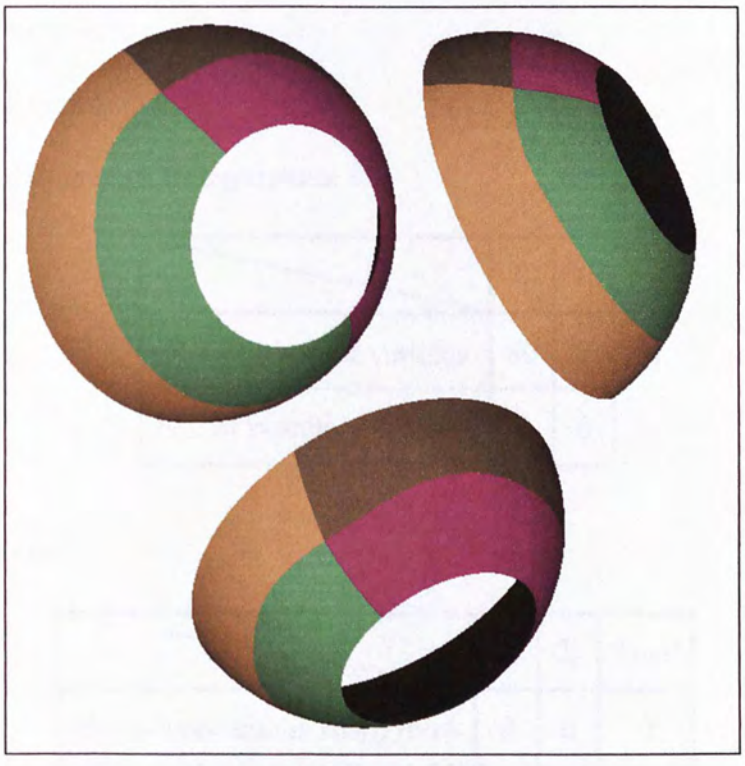

Figure 8.14 Surface patches (3D space)

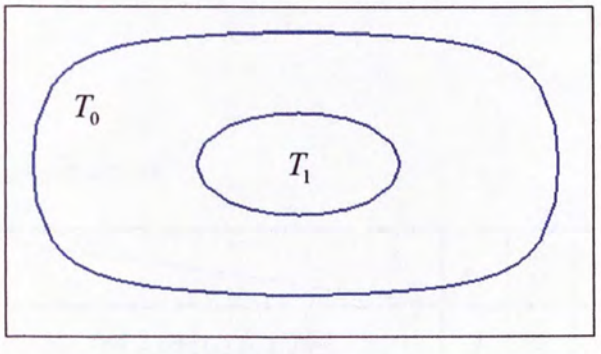

Figure 8.15 Original trimmed surface (parametric space)

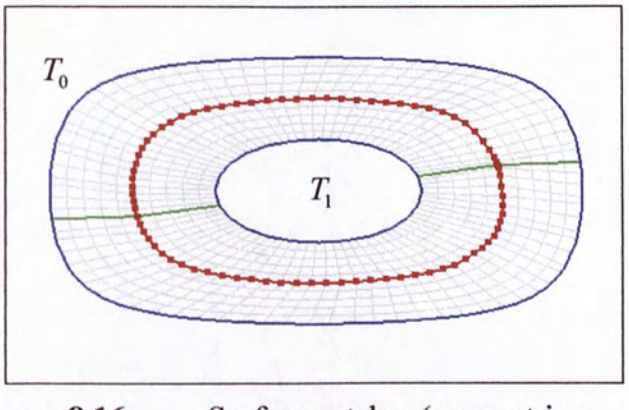

Figure 8.16 Surface patches (parametric space)

Details of Example 4:

1. Voronoï diagram

   - Initial degree of triangulation: 8

	T_0	T_1
No. of bisector vertices	80	80
No. of bisector centroids	0	0

2. Feature points

	T_0	T_1	Total
No. of continuous sharp turns	0	0	0
No. of discrete sharp turns	0	0	0
Total no. of feature points			0

3. Surface patches

   - u-Order: 4, v-Order: 6

	T_0	T_1	Total
No. of surface patches	2	2	4
(No. of degenerated surface patches)	(0)	(0)	(0)

8.5. Example 5: Hemisphere 2

Figure 8.17 Original trimmed surface (3D space)

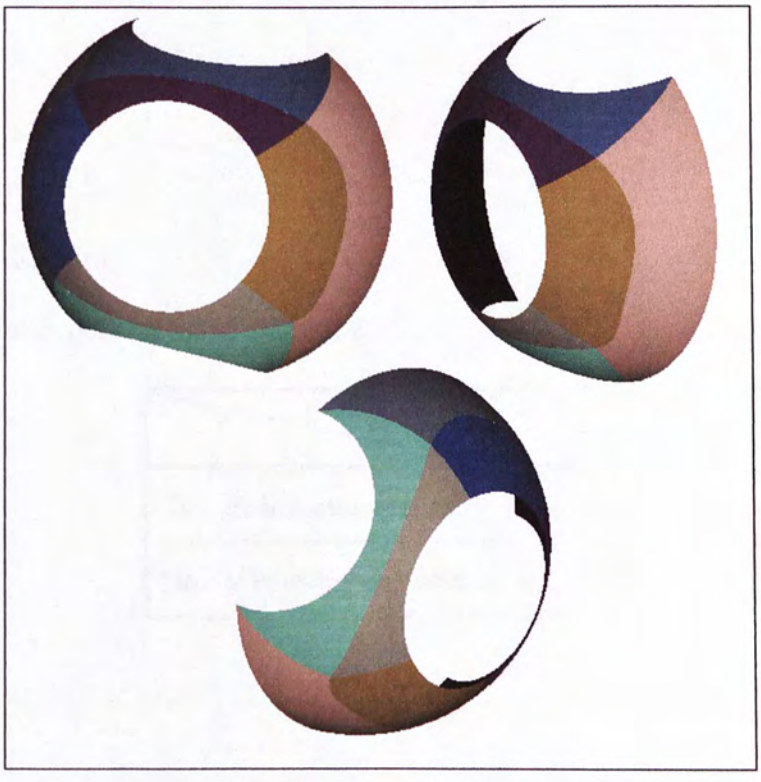

Figure 8.18 Surface patches (3D space)

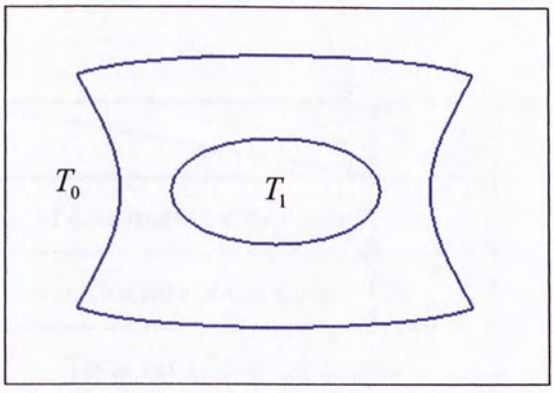

Figure 8.19 Original trimmed surface (parametric space)

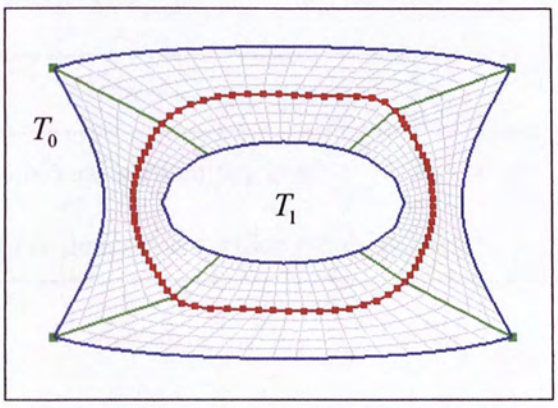

Figure 8.20 Surface patches (parametric space)

Details of Example 5:

1. Voronoï diagram

   - Initial degree of triangulation: 8

	T_0	T_1
No. of bisector vertices	80	80
No. of bisector centroids	0	0

2. Feature points

	T_0	T_1	Total
No. of continuous sharp turns	0	0	0
No. of discrete sharp turns	4	0	4
Total no. of feature points			4

3. Surface patches

- u-Order: 4, v-Order: 6

	T_0	T_1	Total
No. of surface patches	4	4	8
(No. of degenerated surface patches)	(0)	(0)	(0)

8.6. Example 6: Shoe

Figure 8.21 Original trimmed surface (3D space)

Figure 8.22 Surface patches (3D space)

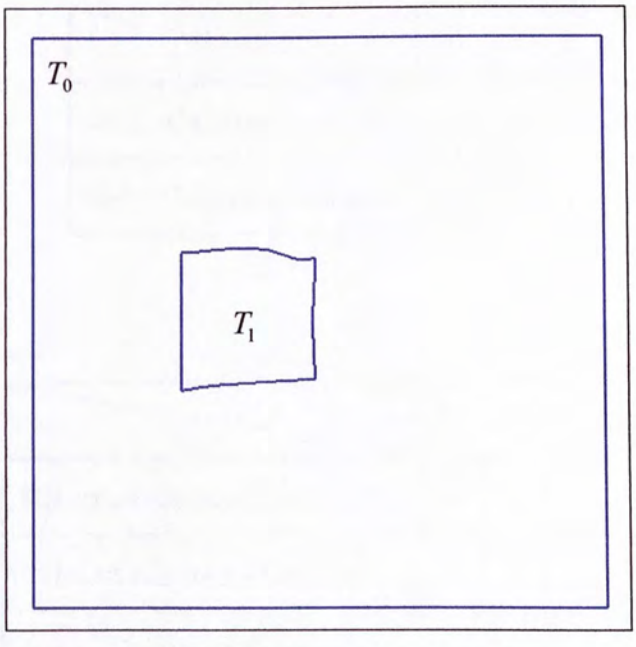

Figure 8.23 Original trimmed surface (parametric space)

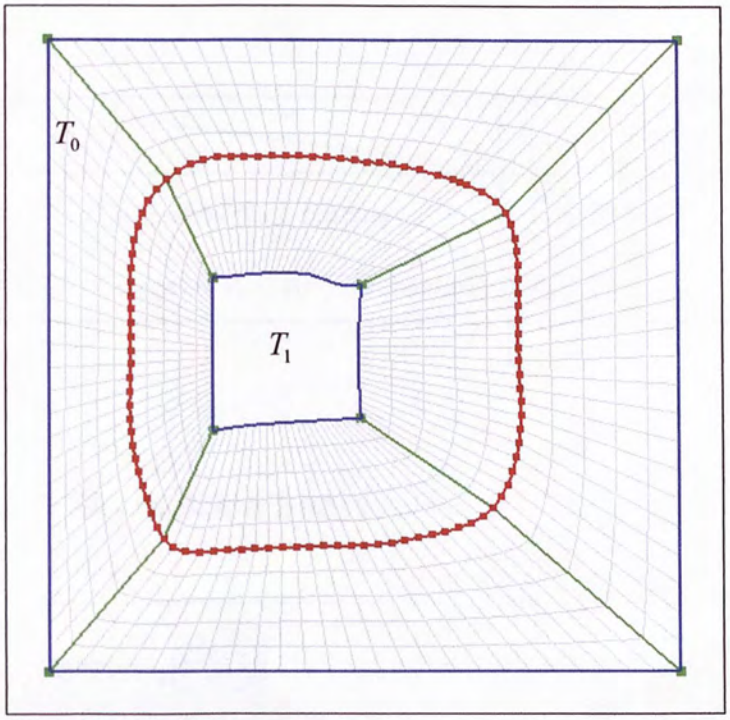

Figure 8.24 Surface patches (parametric space)

Details of Example 6:

1. Voronoï diagram

   - Initial degree of triangulation: 8

	T_0	T_1
No. of bisector vertices	100	100
No. of bisector centroids	0	0

2. Feature points

	T_0	T_1	Total
No. of continuous sharp turns	0	0	0
No. of discrete sharp turns	4	4	8
Total no. of feature points			8

3. Surface patches

	T_0	T_1	Total
No. of surface patches	4	4	8
(No. of degenerated surface patches)	(0)	(0)	(0)

8.7. Example 7: Shark Main Body

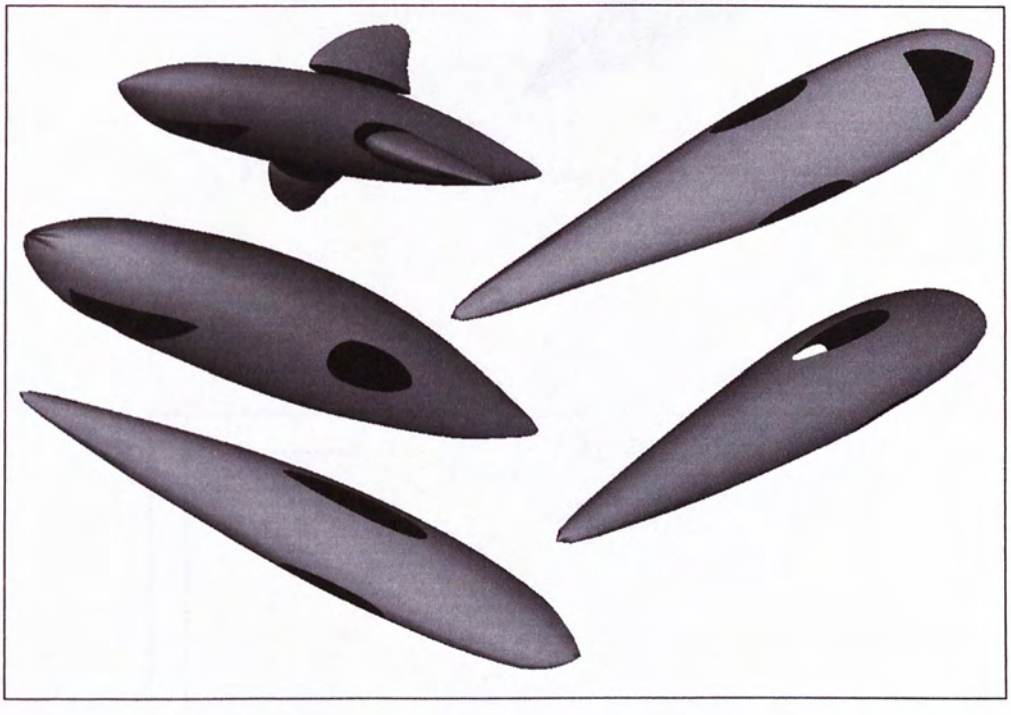

Figure 8.25 Original trimmed surface (3D space)

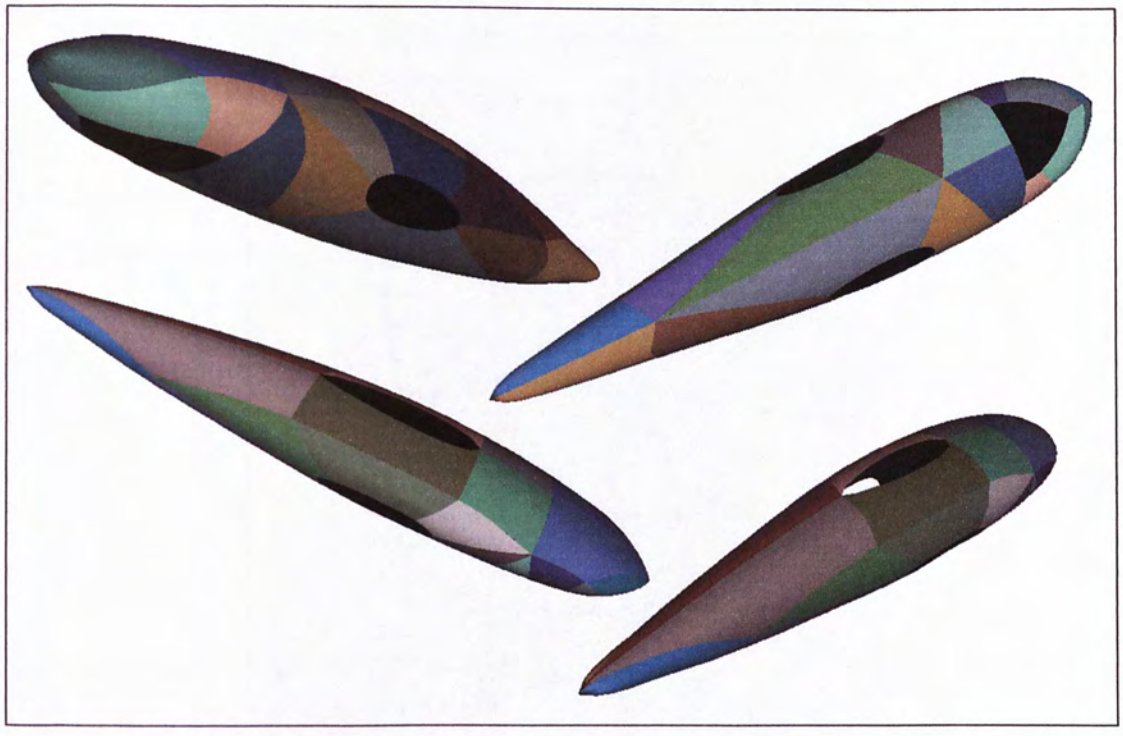

Figure 8.26 Surface patches (3D space)

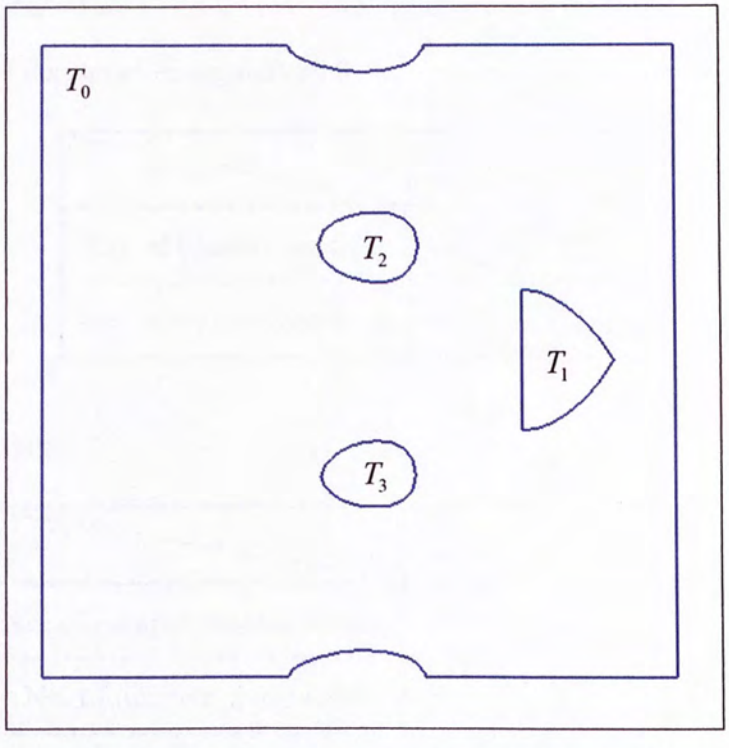

Figure 8.27 Original trimmed surface (parametric space)

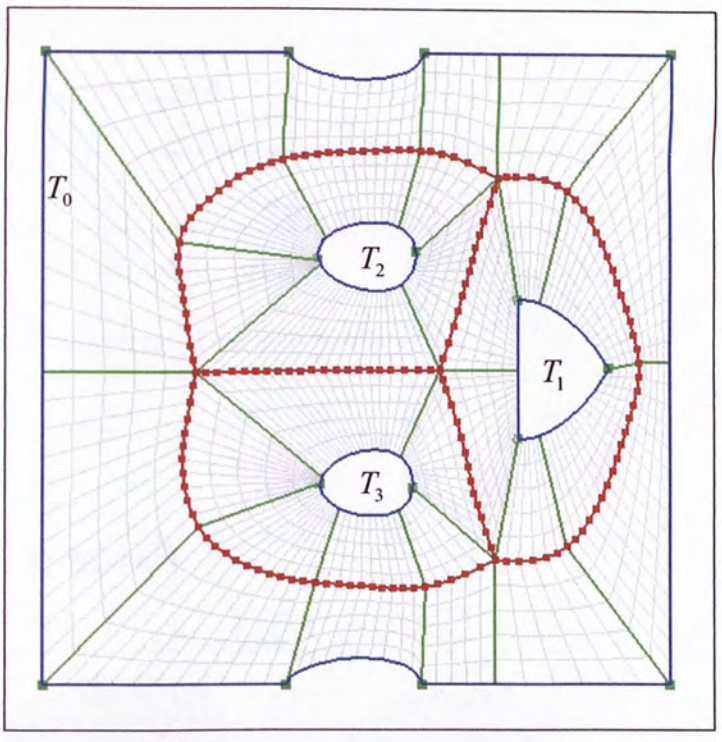

Figure 8.28 Surface patches (parametric space)

Details of Example 7:

1. Voronoï diagram

   - Initial degree of triangulation: 8

	T_0	T_1	T_2	T_3
No. of bisector vertices	116	69	72	73
No. of bisector centroids	3	3	3	3

2. Feature points

	T_0	T_1	T_2	T_3	Total
No. of continuous sharp turns	0	0	2	2	4
No. of discrete sharp turns	8	3	0	0	11
Total no. of feature points					15

3. Surface patches

- u-Order: 4, v-Order: 6

	T_0	T_1	T_2	T_3	Total
No. of surface patches	12	6	6	6	30
(No. of degenerated surface patches)	(0)	(0)	(0)	(0)	(0)

8.8. Example 8: Mask 1

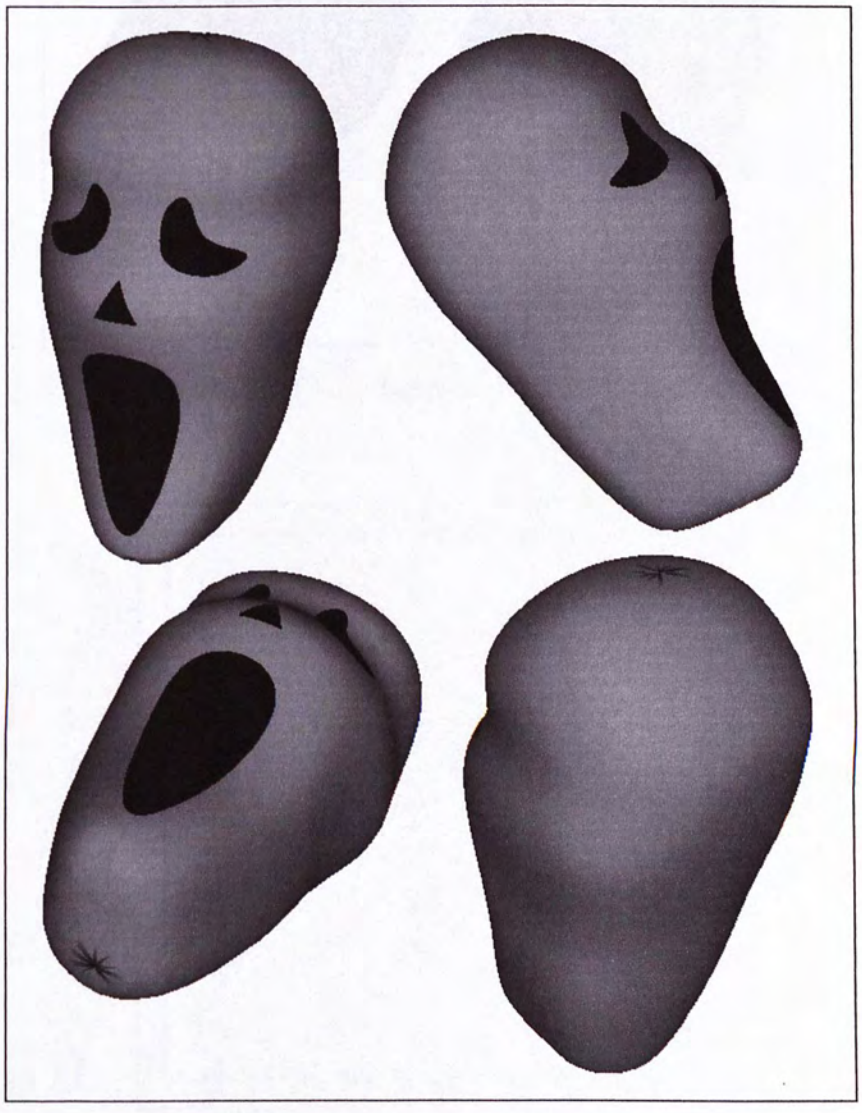

Figure 8.29 Original trimmed surface (3D space)

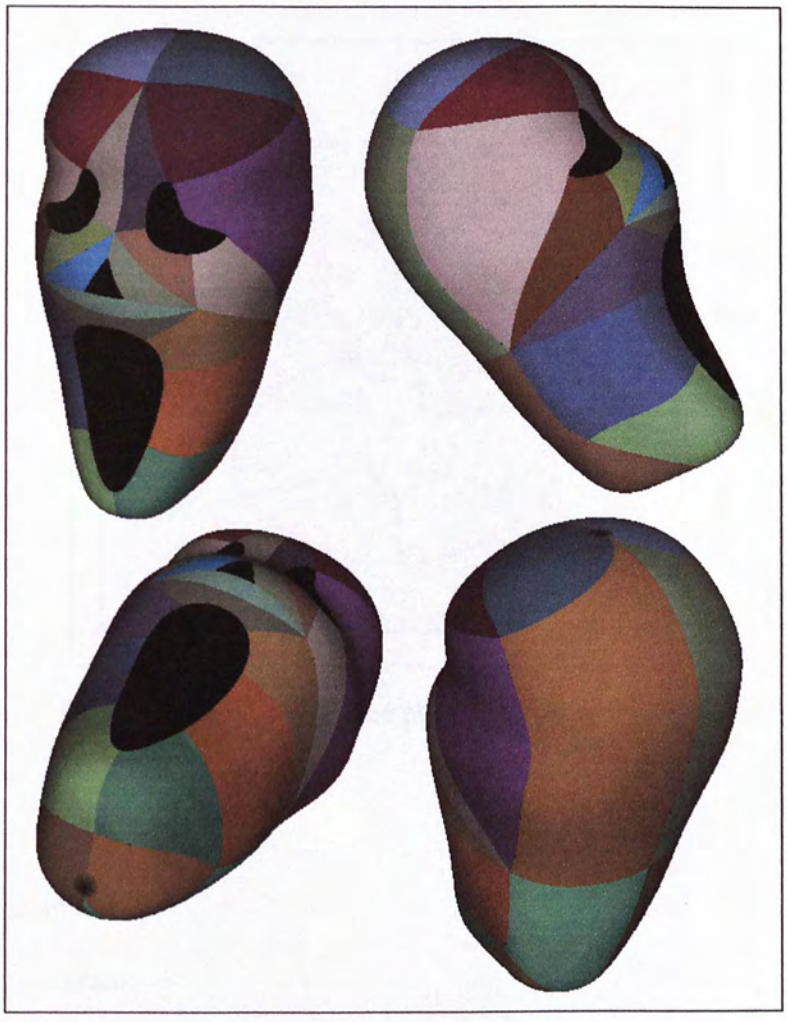

Figure 8.30 Surface patches (3D space)

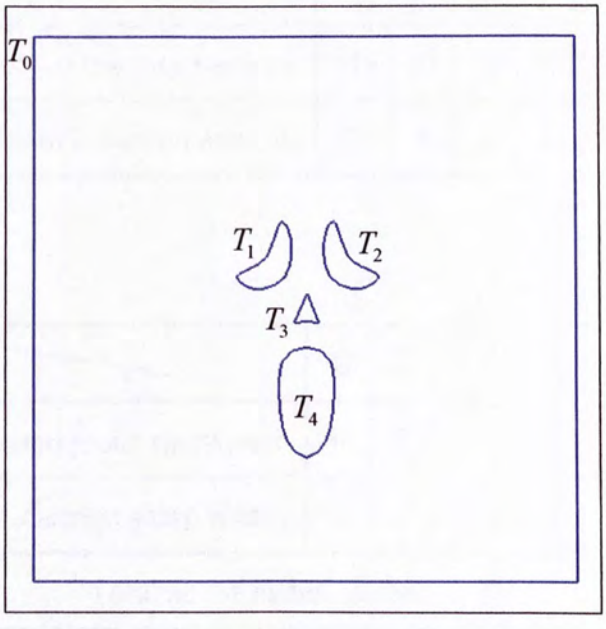

Figure 8.31 Original trimmed surface (parametric space)

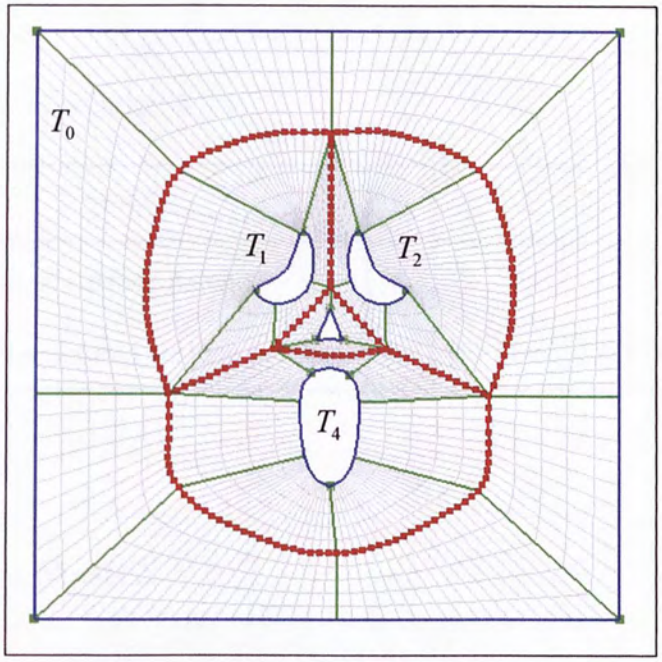

Figure 8.32 Surface patches (parametric space)

Details of Example 8:

1. Voronoï diagram

   - Initial degree of triangulation: 8

	T_0	T_1	T_2	T_3	T_4
No. of bisector vertices	146	82	82	30	96
No. of bisector centroids	3	4	4	3	4

2. Feature points

	T_0	T_1	T_2	T_3	T_4	Total
No. of continuous sharp turns	0	2	2	0	3	7
No. of discrete sharp turns	4	0	0	3	0	7
Total no. of feature points						14

3. Surface patches

- u-Order: 4, v-Order: 6

	T_0	T_1	T_2	T_3	T_4	Total
No. of surface patches	8	5	5	3	7	28
(No. of degenerated surface patches)	(0)	(0)	(0)	(0)	(0)	(0)

8.9. Example 9: Mask 2

Figure 8.33 Original trimmed surface (3D space)

Figure 8.34 Surface patches (3D space)

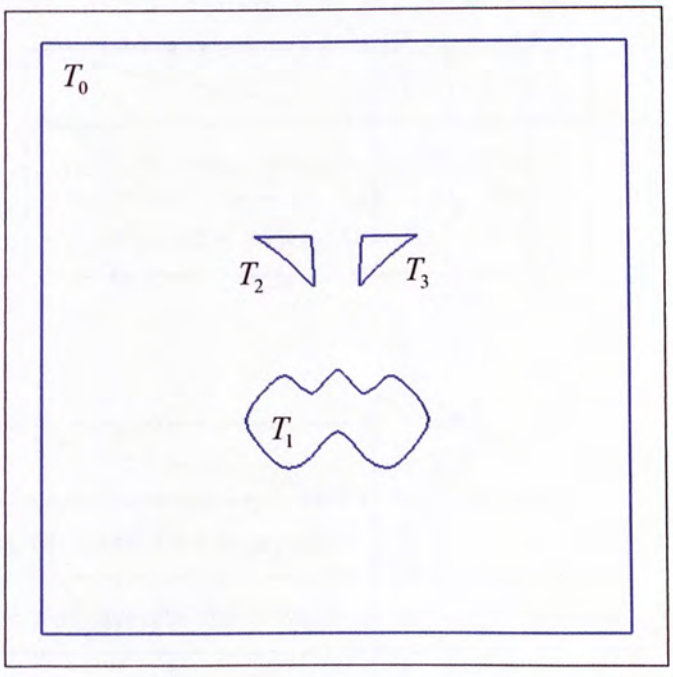

Figure 8.35 Original trimmed surface (parametric space)

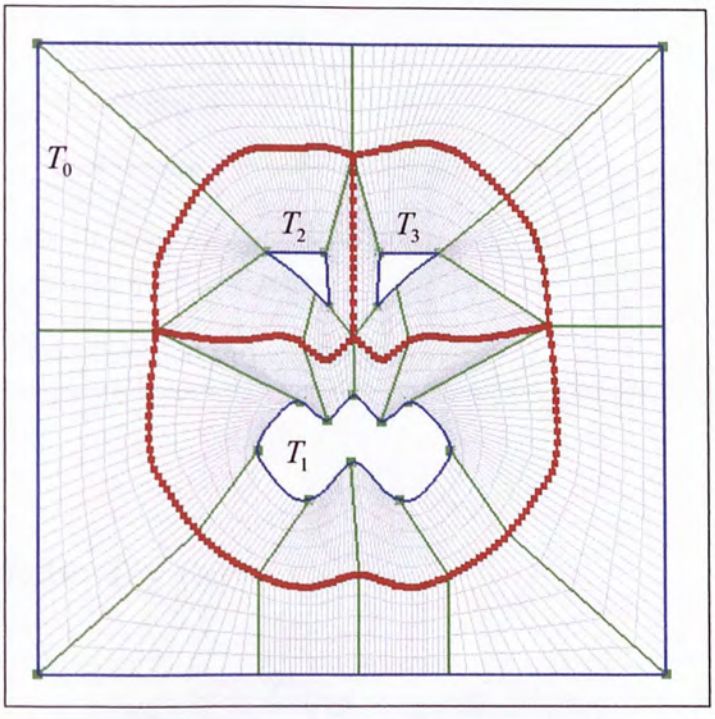

Figure 8.36 Surface patches (parametric space)

Details of Example 9:

1. Voronoï diagram

   - Initial degree of triangulation: 12

	T_0	T_1	T_2	T_3
No. of bisector vertices	148	135	73	74
No. of bisector centroids	3	3	3	3

2. Feature points

	T_0	T_1	T_2	T_3	Total
No. of continuous sharp turns	0	10	0	0	10
No. of discrete sharp turns	4	0	3	3	10
Total no. of feature points					20

Surface patches

- u-Order: 4, v-Order: 6

	T_0	T_1	T_2	T_3	Total
No. of surface patches	10	10	5	5	30
(No. of degenerated surface patches)	(0)	(0)	(1)	(1)	(2)

8.10. Example 10: Toy Car

Figure 8.37 Original trimmed surface (3D space)

Figure 8.38 Surface patches (3D space)

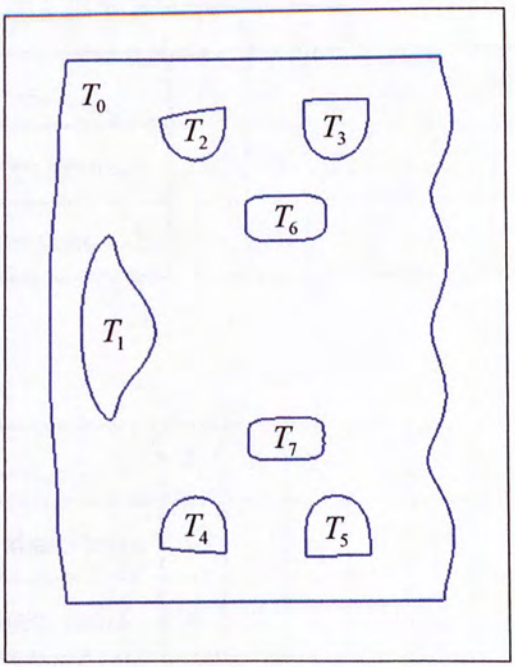

Figure 8.39 Original trimmed surface (parametric space)

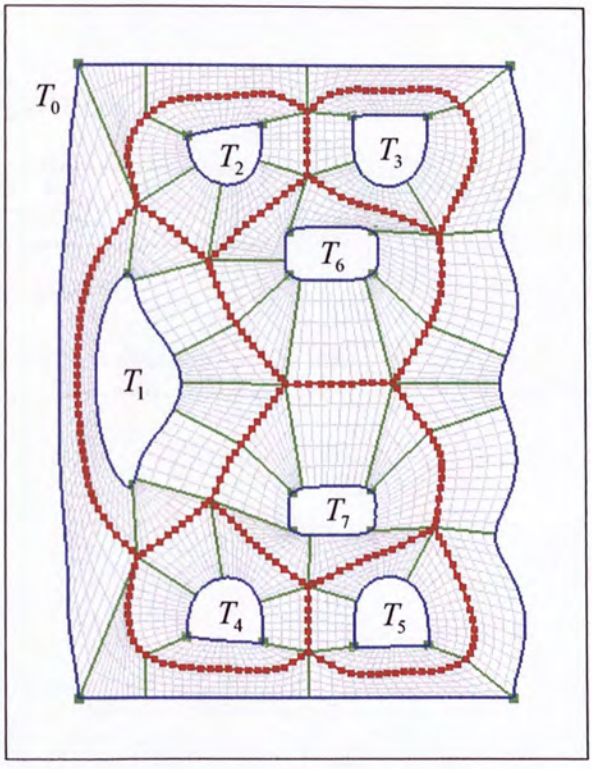

Figure 8.40 Surface patches (parametric space)

Details of Example 10:

1. Voronoï diagram

   - Initial degree of triangulation: 8

	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7
No. of bisector vertices	195	88	59	56	52	54	70	70
No. of bisector centroids	7	5	4	3	4	3	5	5

2. Feature points

	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	Total
No. of continuous sharp turns	0	2	0	0	0	0	4	4	10
No. of discrete sharp turns	4	0	2	2	2	2	0	0	12
Total no. of feature points									22

3. Surface patches

- u-Order: 4, v-Order: 6

	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	Total
No. of surface patches	13	7	5	5	4	4	7	7	52
(No. of degenerated surface patches)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)	(0)

Chapter 9. Result and Analysis

In this chapter, some issues related to the continuity and degeneration of the decomposed patches are discussed. A comparison between the proposed algorithm and Hamann's algorithm will also be presented.

9.1. Continuity between Patches

In the establishment of correspondence and the construction of correspondence links, adjacent parametric patches share the same set of data points on their common edges. This applies to both the *u*- and *v*-directions of the parametric patches, i.e., along the bisector segments and the correspondence links.

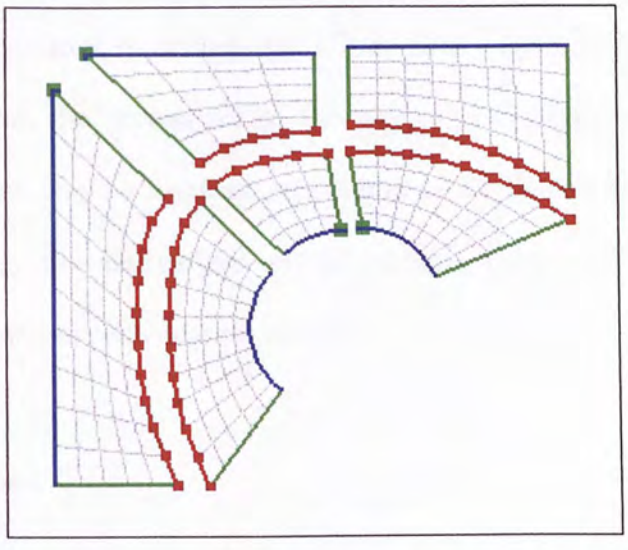

Figure 9.1 Neighbouring parametric patches

A graphical interpretation is shown in *Figure 9.1*. The example in the figure is a set of neighbouring parametric patches which share common edges on their boundaries. Patches sharing the same edges share the same set of data points, in the

parametric and thus the 3D space. Provided the same knot sequence is used and the data points are assigned the same parametric values, C^0 continuity can be maintained.

Although the generated patches may not be C^1 continuous, it is obvious that if more data points are sampled for the surface fitting process, C^1 continuity between the patches can be approached. This is because a larger number of sample points give a better approximation of the original surface which is C^1 continuous. This can be achieved by increasing the number of data points sampled in the local u- and v-directions of the patches respectively. To increase the data points along the u-direction, more bisector vertices are generated on the bisector segment of every parametric patch. Similarly, along the v-direction, the number of data points can be increased by increasing the points sampled on the correspondence links.

In case it is desired to ensure the C^1 or even higher degrees of continuity between the patches, the derivatives of the surface at the sample points along the common boundaries can be used as constraints in the interpolation process. This requires estimating the derivatives at the sample points and requires special consideration for patches with degenerate edge.

9.2. Special Cases

In the establishment of vertices correspondence, if two or more bisector vertices are associated with the same data point on the trim, a triangular patch with a degenerated edge will be obtained. In addition, there may exist S-shaped features on the trim shapes which may lead to undesirable results in the surface fitting process as discussed below.

9.2.1. Degenerated Patches

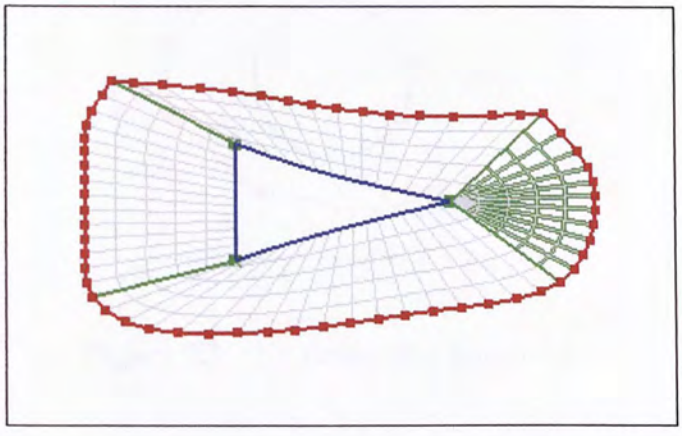

Figure 9.2 Degenerated patch

A degenerated patch is a patch with one edge zero length. The shape of a degenerated patch is triangular in most cases. The highlighted patch in *Figure 9.2* is a typical example.

To avoid the generation of degenerated patches in a parametric tile, a bisector vertex is not allowed to be associated with more than one data point or feature point on the trim in the forward attachment process. In *Figure 9.3*, the solid green line represents the correspondence which is built between a feature point and a bisector vertex in the forward attachment process. In the subsequent process, any feature point attempting to associate with the same bisector vertex (e.g. the dashed green line in the figure) will not be allowed.

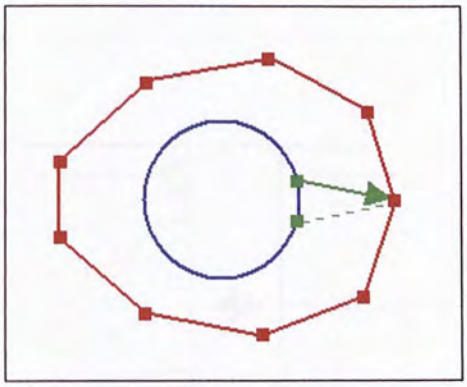

Figure 9.3 Preventing degeneration

However, the same technique cannot be applied in the backward attachment process, i.e., the process of associating an un-associated bisector centroid to its closest data point on the trim. This is because of the possibility of generating patches with irregular shape such as concave boundary as shown in *Figure 9.4*. The dashed line in *Figure 9.4* represents the link which is built while degeneracy is allowed in the backward attachment process. The solid line directed from the same bisector vertex represents the link built while degeneracy is not allowed. It can be seen that the patch without degeneracy is concave in shape. In this case, a degenerated patch is more preferable than a concave patch. Therefore, bisector vertices are allowed to associate with the same feature point in the backward attachment process.

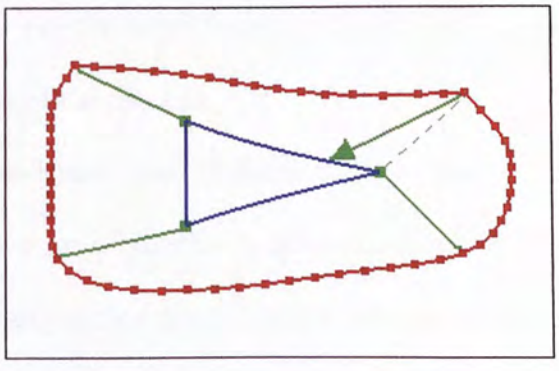

Figure 9.4 Concave patch

9.2.2. S-Shaped Feature

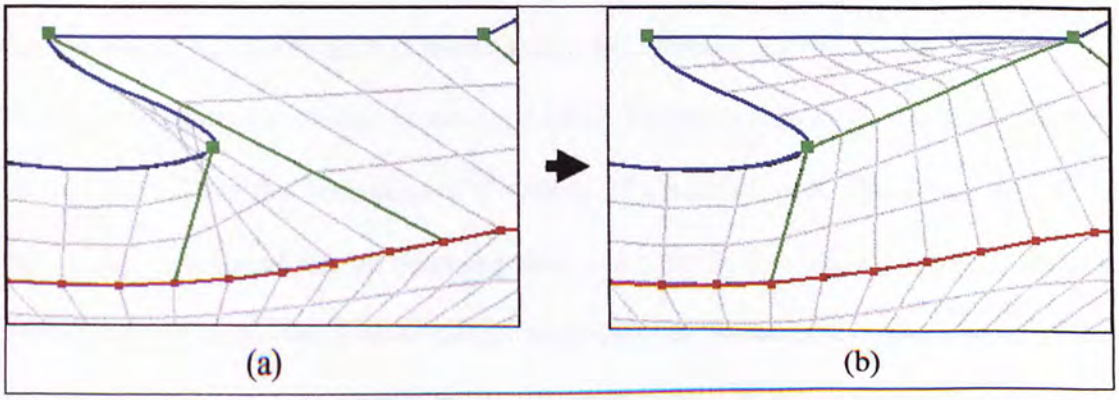

Figure 9.5 Solution to S-shaped feature

S-shaped features on the trim shape may cause the parametric grid lines generated inside a patch to intersect with each other and the trim segment. The corresponding surface patch is twisted (*Figure 9.5a*). This can be avoided by rearranging the correspondence links and making use of degenerated patches *Figure 9.5b*.

9.3. Comparison

In this section, the feature-based algorithm presented in this thesis will be compared with the algorithm developed by Hamann and Tsai. The comparison is based on several aspects. They are the total number of patches, the number of degenerated patches and the shapes of the patches.

Both the feature-based and Hamann's algorithm effectively decompose a trimmed surface into a set of regular B-spline surfaces. However, using Hamann's approach, the number of patches developed depends on the number of local maximum and minimum points on the bisectors and trimming curves. The number of patches generated by the feature-based approach depends on the number of the feature points

on the trims and also the number of bisector centroids on the bisectors. The maximum number of patches is of the order $O(n^t + Nn^{bc})$, where n^t is the total number of feature points on the trims, N is the total number of trims, and n^{bc} is the total number of bisector centroids on the bisectors. Unlike Hamman's method, the feature-based method considers the boundary of a surface as a trim as well. Therefore, one more bisector is developed for the boundary trim. As a result, one more parametric tile has to be processed in the feature-based algorithm. In most cases, fewer patches are usually obtained by using Hamann's method. However, there are exceptional cases as shown in *Figure 9.6*, when the number of patches obtained with the feature-based approach is less than that obtained with Hamann's method. This usually occurs when the number of extreme points is larger than the number of feature points. *Figure 9.6a* and *Figure 9.6b* shows the results of decomposing a surface using respectively Hamann's and the feature-based method. In the figure, twelve patches are obtained by Hamann's algorithm, while eight patches are obtained by the feature-based algorithm.

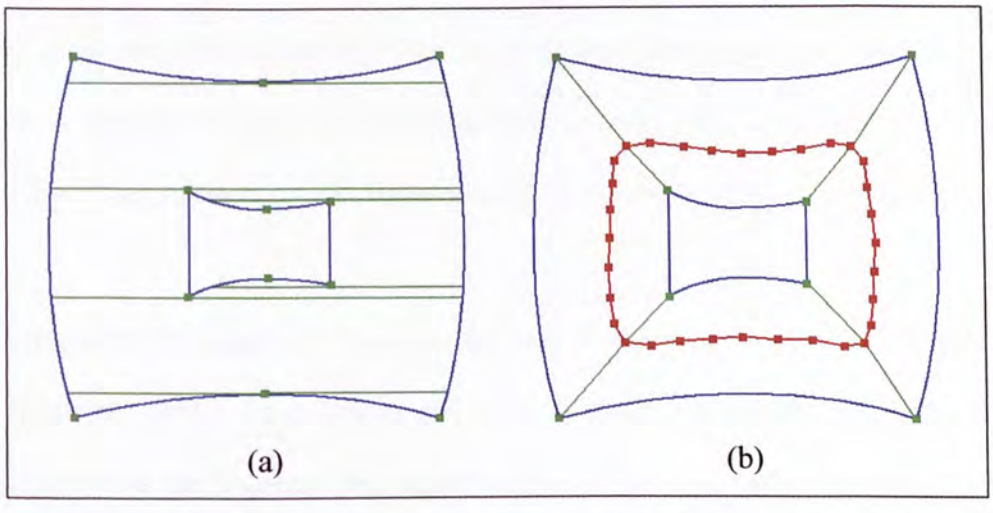

Figure 9.6 Comparison between (a) Hamann's and (b) feature-based algorithms

In the feature-based algorithm, the shapes of the patches created depend on the shapes of the trims and the bisectors. They also depend on the similarity between the shape of the trims and their bisector loops. In general, the patches are four-sided. The number of degenerated patches can be reduced by not allowing two or more feature points to associate with the same bisector vertex. Moreover, the patches with concave shapes can be avoided. The shapes of the patches created using Hamann's method depend on the distribution of the maximum and minimum points on the trimming curves. Patches are degenerated wherever there is a single pair of adjacent maximum and minimum points, as shown in *Figure 9.6a*.

By applying the technique of shape blending, the feature-based algorithm establishes vertices correspondence by considering the shapes of the trims and the bisector loops. The shapes of the patches are thus more regular. For Hamann's method, narrow patches may be obtained as shown in *Figure 9.6a*. This is caused by closeness of the scan lines passing through the maximum and minimum points. These scan lines are close to each other but are not close enough to be regarded a single scan line. Narrow stripes patches will thus be generated. These kinds of patches usually have their shapes and area very different from the other patches. It can be concluded that the patches created by the feature-based method are more regular in shape and size.

Figure 9.7 to *Figure 9.16* shows the ten examples in Chapter 8 developed by both Hamann's method and the feature-based method in part (a) and (b) of the figures respectively. In the figures, the degenerated patches generated in each method is coloured in grey. The comparison is mainly based on:

1. The total number of surface patches generated
2. The number of degenerated surface patches
3. The area of the parametric patches

The results of the comparison are listed in the table below the figure of each example. For the area comparison, the data is expressed in the percentage between the area of a specific patch and the total area of all patches. For example, the data "Max." is the percentage of the patch with maximum area compared with the total area of all patches.

By observing the statistics of the comparison, the number of patches generated by Hamann's method is smaller than that by the feature-based method. However, regarding the number of degenerated patches, the feature-based method has an advantage over Hamann's method. The area variation of the patches obtained by the feature-based method is smaller too. This can be proved by the difference between the maximum and minimum patches areas, and also the areas standard deviation (S.D.) in the tables. The shapes and regularity of the patches generated by both methods can also be compared by observation.

In the examples, the patch areas are measure by a free image processing and analysis program called *ImageTool* [12]. The measurement is done by defining a region in an image file and the area is equal to the total number of pixel in that region.

9.3.1. Example 1: Deformed Plane 1

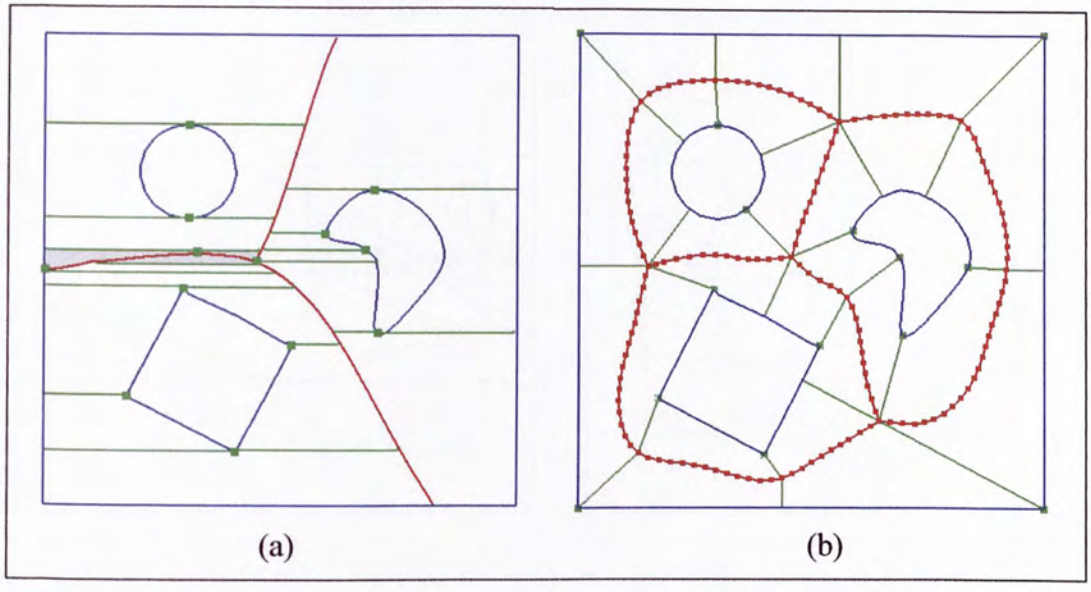

Figure 9.7 Example 1 by (a) Hamann's and (b) feature-based method

		Hamann's	Feature-based
No. of surface patches		22	26
No. of degenerated surface patches		3	0
No. of extreme/feature points		13	14
$\frac{\text{Area of patch}}{\text{Total area of patches}}$	Max.	16.43%	8.72%
	Min.	0.19%	1.59%
	Mean	4.55%	3.85%
	S.D.	4.55%	1.83%

9.3.2. Example 2: Deformed Plane 2

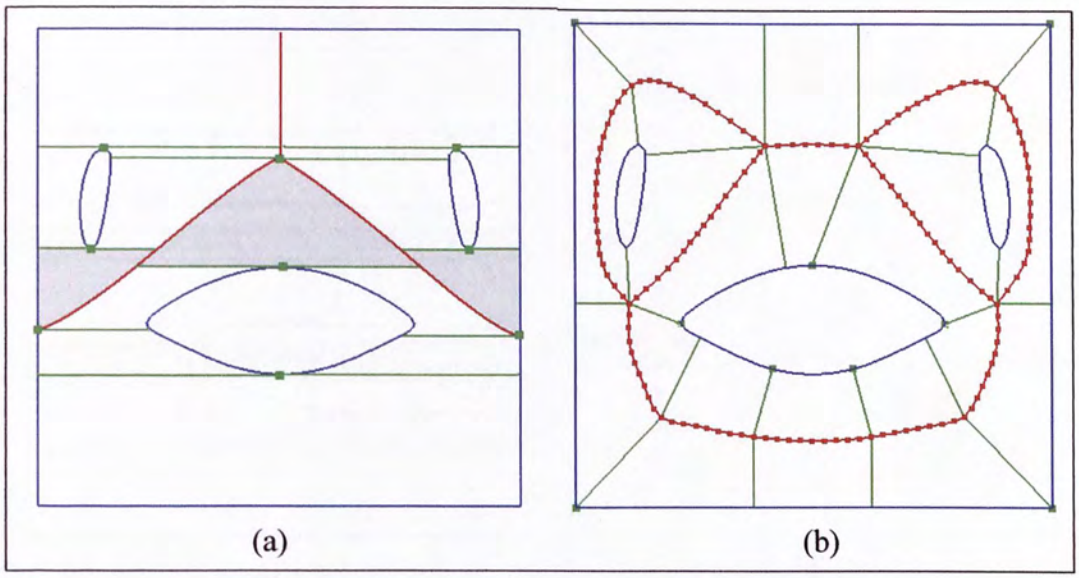

Figure 9.8 Example 2 by (a) Hamann's and (b) feature-based method

		Hamann's	Feature-based
No. of surface patches		16	24
No. of degenerated surface patches		3	0
No. of extreme/feature points		9	9
$\frac{\text{Area of patch}}{\text{Total area of patches}}$	Max.	30.33%	6.12%
	Min.	0.84%	2.16%
	Mean	6.25%	4.17%
	S.D.	7.52%	1.33%

9.3.3. Example 3: Sphere

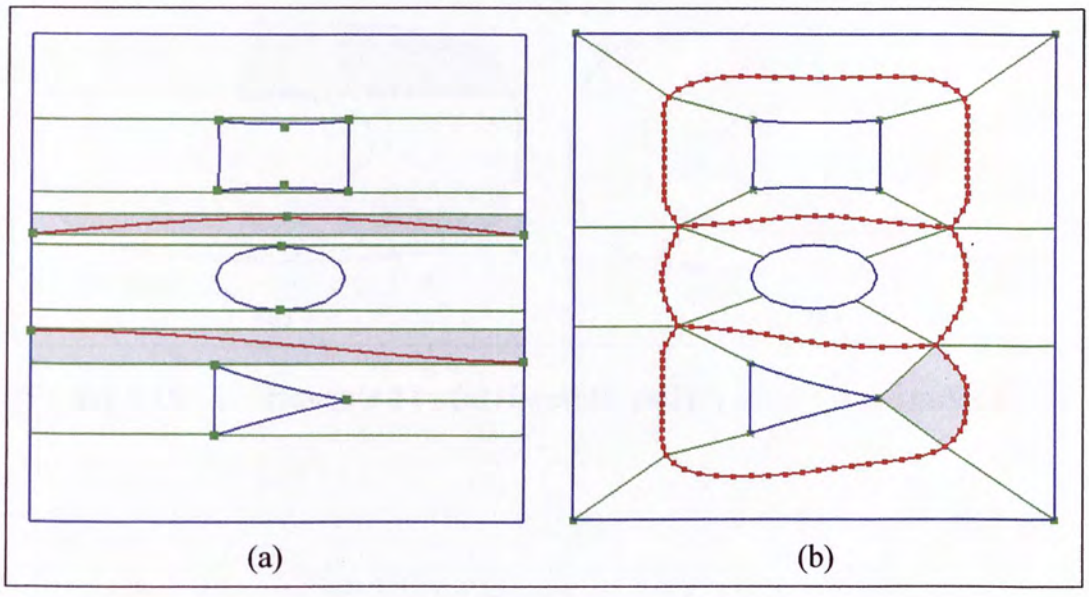

Figure 9.9 Example 3 by (a) Hamann's and (b) feature-based method

		Hamann's	Feature-based
No. of surface patches		19	20
No. of degenerated surface patches		5	1
No. of extreme/feature points		16	11
$\frac{\text{Area of patch}}{\text{Total area of patches}}$	Max.	19.49%	9.91%
	Min.	0.17%	2.37%
	Mean	5.26%	5%
	S.D.	5.28%	2.05%

9.3.4. Example 4: Hemisphere 1

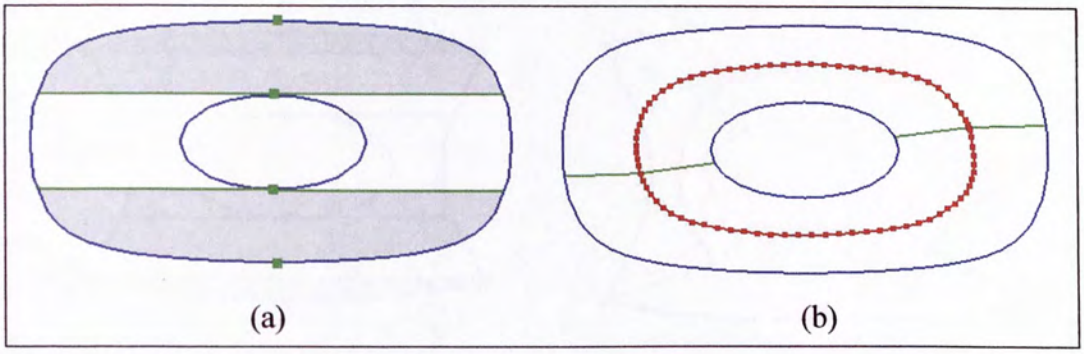

Figure 9.10 Example 4 by (a) Hamann's and (b) feature-based method

		Hamann's	Feature-based
No. of surface patches		4	4
No. of degenerated surface patches		2	0
No. of extreme/feature points		4	0
$\frac{\text{Area of patch}}{\text{Total area of patches}}$	Max.	31.90%	31.14%
	Min.	18.22%	18.90%
	Mean	25%	25%
	S.D.	7.83%	6.92%

9.3.5. Example 5: Hemisphere 2

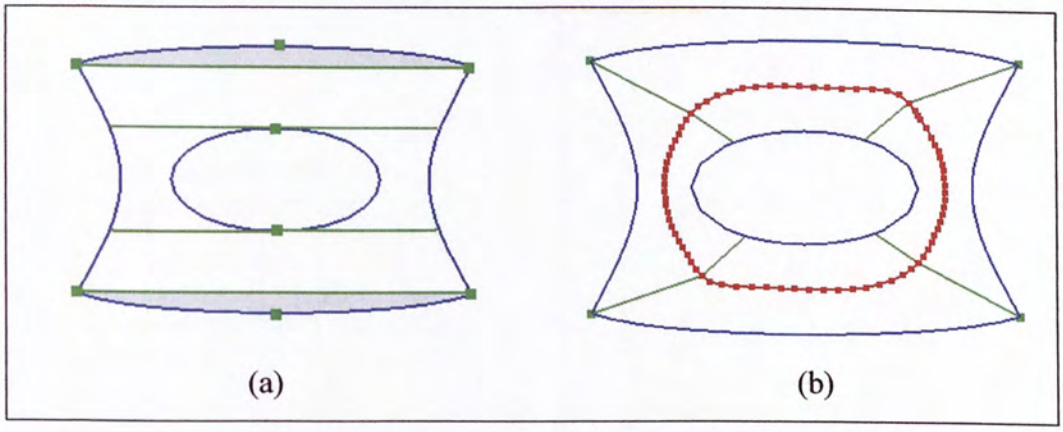

Figure 9.11 Example 5 by (a) Hamann's and (b) feature-based method

		Hamann's	Feature-based
No. of surface patches		6	8
No. of degenerated surface patches		2	0
No. of extreme/feature points		8	4
$\frac{\text{Area of patch}}{\text{Total area of patches}}$	Max.	30.91%	19.93%
	Min.	6.92%	9.94%
	Mean	16.67%	12.5%
	S.D.	10.95%	5.07%

9.3.6. Example 6: Shoe

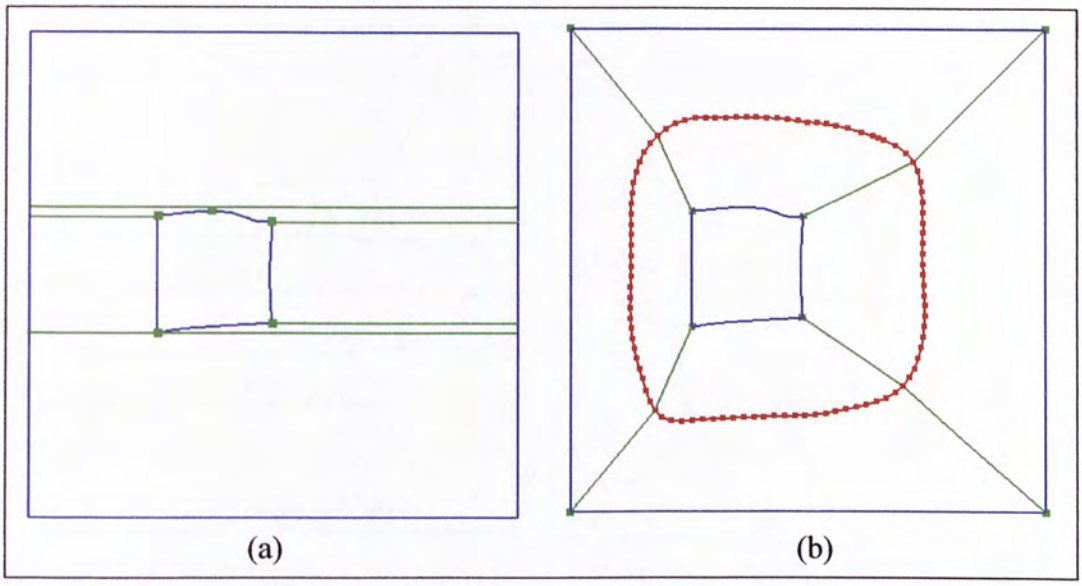

Figure 9.12 Example 6 by (a) Hamann's and (b) feature-based method

		Hamann's	Feature-based
No. of surface patches		7	8
No. of degenerated surface patches		0	0
No. of extreme/feature points		5	8
Area of patch / Total area of patches	Max.	40.16%	20.69%
	Min.	0.59%	5.78%
	Mean	14.29%	12.5%
	S.D.	17.35%	5.58%

9.3.7. Example 7: Shark Main Body

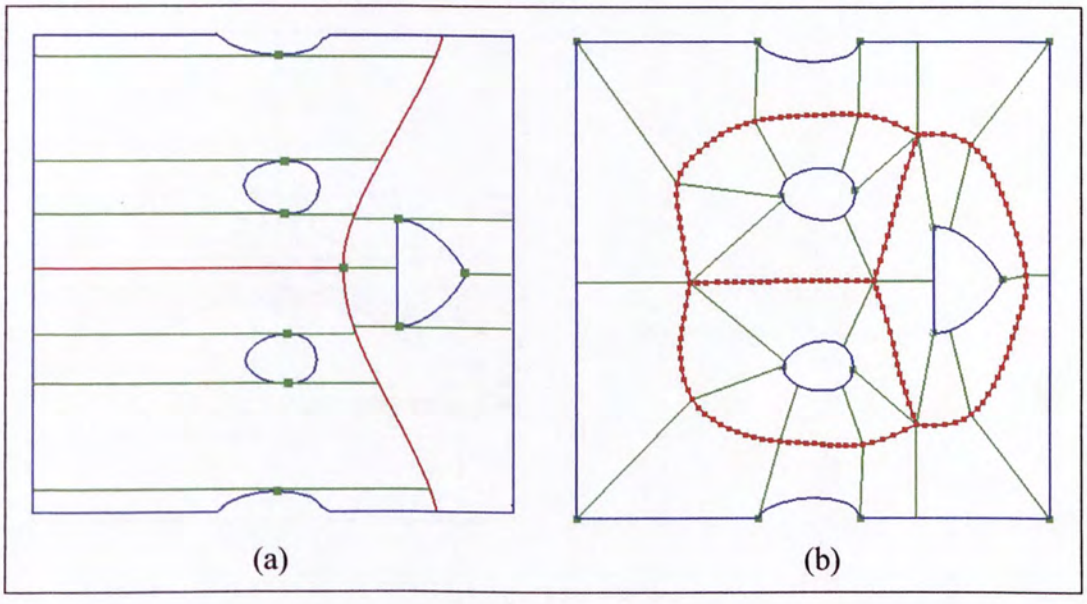

Figure 9.13 Example 7 by (a) Hamann's and (b) feature-based method

		Hamann's	Feature-based
No. of surface patches		18	30
No. of degenerated surface patches		0	0
No. of extreme/feature points		10	15
Area of patch / Total area of patches	Max.	18.60%	9.02%
	Min.	1.17%	1.24%
	Mean	5.55%	3.33%
	S.D.	5.89%	2.04%

9.3.8. Example 8: Mask 1

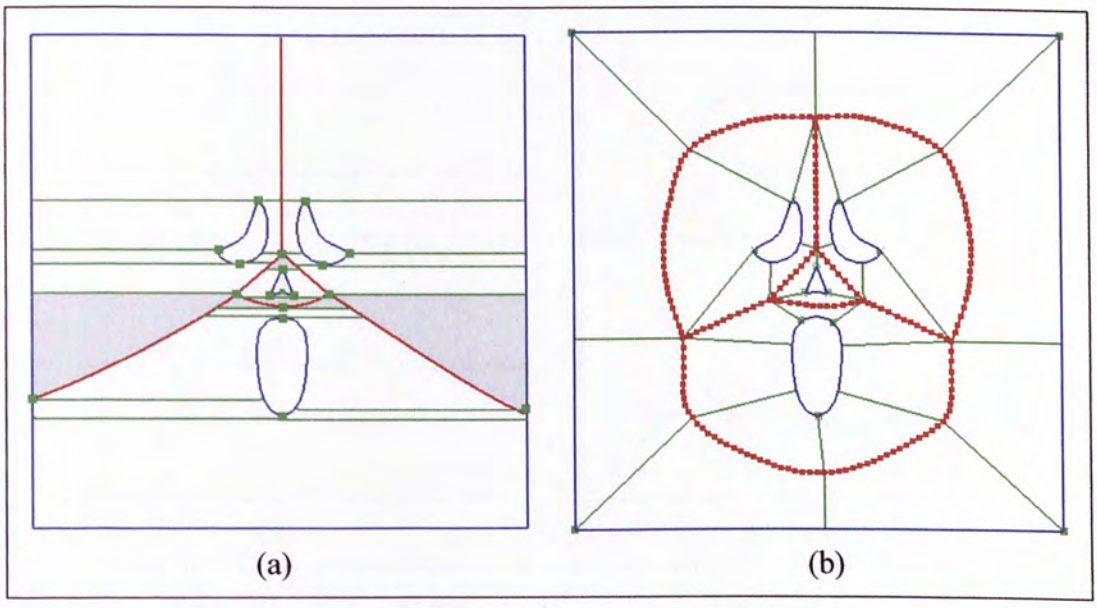

Figure 9.14 Example 8 by (a) Hamann's and (b) feature-based method

		Hamann's	Feature-based
No. of surface patches		27	28
No. of degenerated surface patches		6	0
No. of extreme/feature points		18	14
$\frac{\text{Area of patch}}{\text{Total area of patches}}$	Max.	22.53%	10.78%
	Min.	0.04%	0.36%
	Mean	3.70%	3.57%
	S.D.	5.87%	3.27%

9.3.9. Example 9: Mask 2

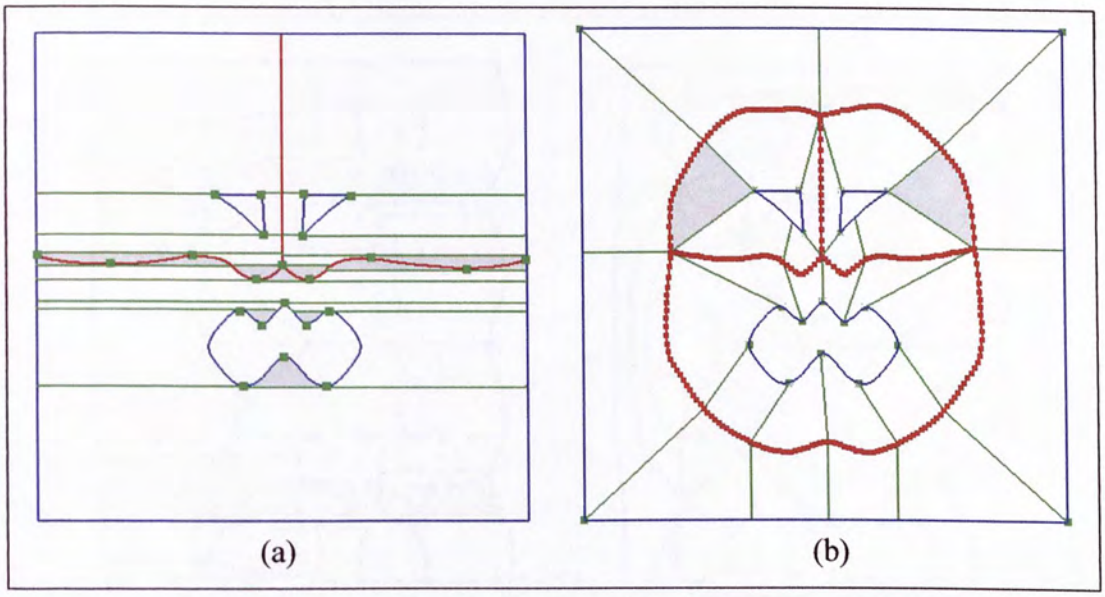

Figure 9.15 Example 9 by (a) Hamann's and (b) feature-based method

		Hamann's	Feature-based
No. of surface patches		30	30
No. of degenerated surface patches		12	2
No. of extreme/feature points		23	20
$\frac{\text{Area of patch}}{\text{Total area of patches}}$	Max.	28.78%	9.42%
	Min.	0.11%	0.57%
	Mean	3.33%	3.33%
	S.D.	6.46%	2.74%

9.3.10. Example 10: Toy Car

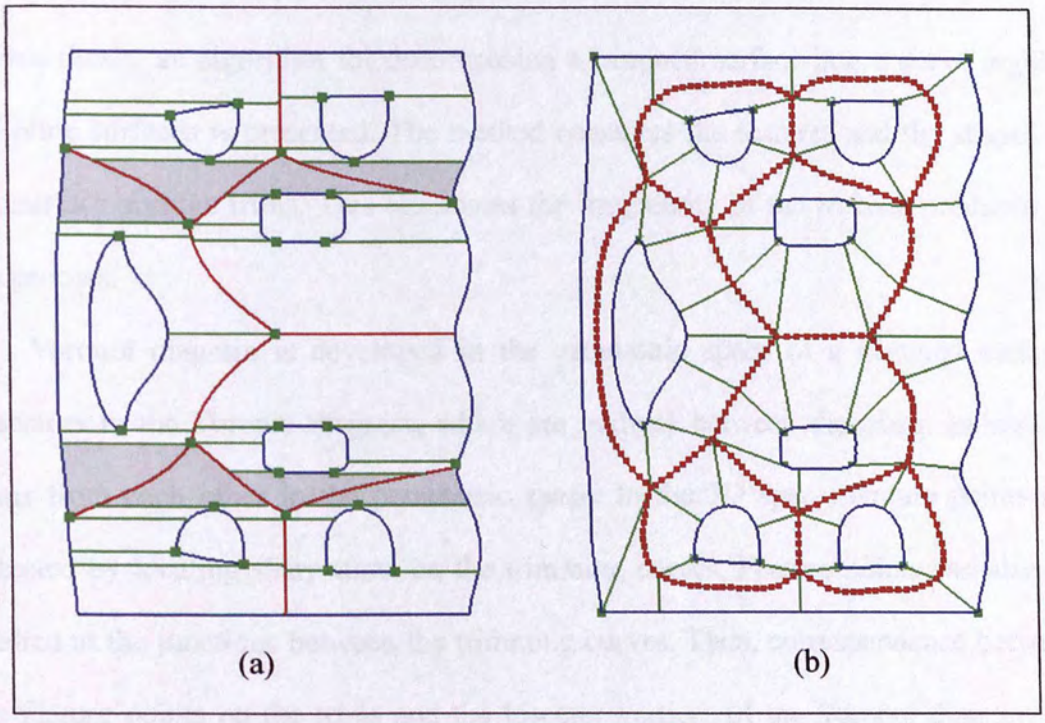

Figure 9.16 Example 10 by (a) Hamann's and (b) feature-based method

		Hamann's	Feature-based
No. of surface patches		43	52
No. of degenerated surface patches		8	0
No. of extreme/feature points		31	22
Area of patch / Total area of patches	Max.	11.16%	8.61%
	Min.	0.37%	0.95%
	Mean	2.33%	1.92%
	S.D.	2.38%	1.18%

Chapter 10. Conclusion

In this thesis, an algorithm for decomposing a trimmed surface into a set of regular B-Spline surfaces is presented. The method considers the features and the shapes of the surface and the trims. This eliminates the irregularity of the patches produced in the process.

Voronoï diagram is developed in the parametric space of a trimmed surface. Bisectors in the Voronoï diagram, which are midway between the trims, isolate the trims from each other in the parametric space. In the 3D space, feature points are detected by locating sharp turns on the trimming curves. Feature points can also be located at the junctions between the trimming curves. Then, correspondence between the feature points on the trims and the bisector vertices of the Voronoï diagram are established in the parametric space. By considering the similarity between the trims and their bisector loops, correspondence between feature points and bisector vertices is established by applying the shape matching technique in shape blending. Connecting feature points and the corresponding bisector vertices partitions the valid region between the trims and the bisectors into regions for the surface fitting process. A set of surface points related to points lying in each of the parametric regions is generated. A B-Spline surface interpolating the surface points in each parametric region is constructed. The result is a set of surface patches of which union approximates the original trimmed surface.

A series of examples and results are shown demonstrating the performance of the algorithm. Special cases are discussed and the solutions to the problems are proposed. Comparing with the Hamann's approach, the feature-based method generates larger number of patches in most cases. However, the feature-based method has the advantages that the patches obtained are more regular in shape and size.

There are rooms for the algorithm to improve. As the result of the algorithm is a set of surface patches in 3D space, the decomposition algorithm should be performed in 3D space as far as possible. This will increase the accuracy of the decomposition and the regularity of the patches obtained. For example, Voronoï diagram can be developed on the surface so that the bisectors divide the valid region between the trims more evenly in 3D space (*Figure 10.1*). This is because the distance between the trims in the parametric and the 3D space maybe different such that bisectors in the parametric space may not be bisectors in the 3D space.

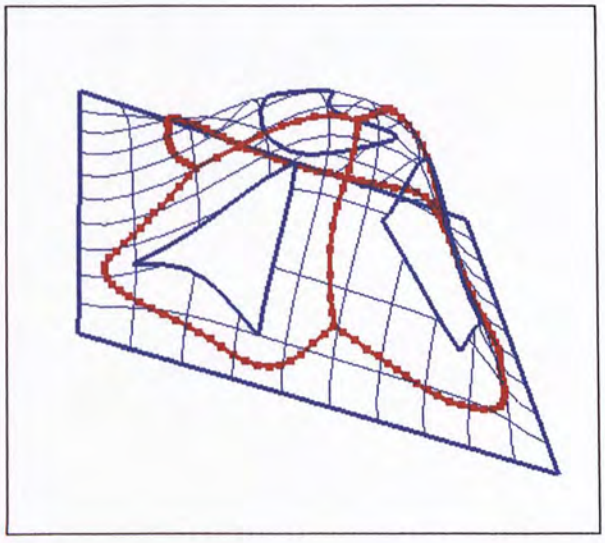

Figure 10.1 Voronoï diagram in 3D space

In the establishment of vertices correspondence, the distance between the feature points and the bisector vertices is measured in parametric space. Measuring the distance in 3D space will give a more accurate result. Besides, in the normalization of the trims and the bisector loops, the actual shapes of the trims and the bisector loops in 3D space can be used. This requires the use of three-dimensional enclosing boxes of the shapes in the normalization process (*Figure 10.2*).

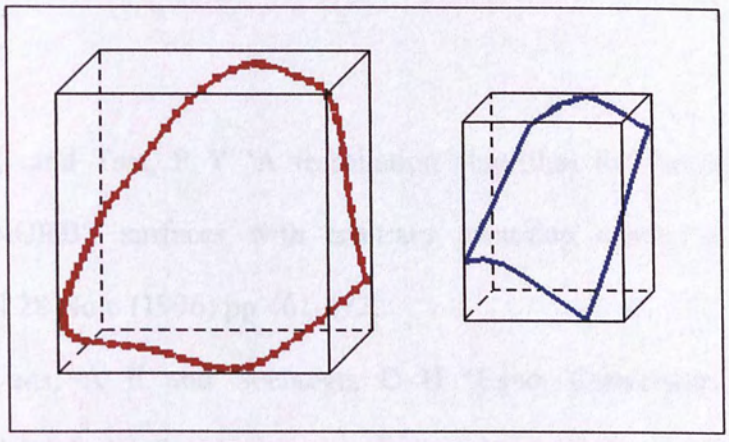

Figure 10.2 Enclosing boxes for normalization in 3D space

References

1. Hamann, B and Tsai, P Y 'A tessellation algorithm for the representation of trimmed NURBS surfaces with arbitrary trimming curves' *Computer-Aided Design* Vol 28 No 6 (1996) pp 461-472.
2. Vries-Baayens, A E and Seebregts C H 'Exact Conversion of a trimmed nonrational Bézier surface into composite or basic nonrational Bézier surfaces' *in* Hagan, H (Ed.) *Topics in Surface Modeling* SIAM, Philadelphia, PA (1992) pp 115-143.
3. Piegl, L A and Richard, A M 'Tessellating trimmed NURBS surfaces' *Computer-Aided Design* Vol 27 No1 (1995) pp 16-26.
4. Cho, W, Maekawa, T, Patrikalakis, N M and Peraire, J 'Robust tessellation of trimmed rational B-spline surface patches' *Proceedings of Computer Graphics International CGI '98* (1998) pp 543-555.
5. Abi-Ezzi, S S and Subramaniam, S 'Fast dynamic tessellation of trimmed NURBS surfaces' *Eurographics '94* Vol 13 No 3 (1994) pp 107-126.
6. Liu, D, Dong, J and Tong R 'A new approach for tessellating trimmed parametric surfaces' *The Fifth International Conference for Young Computer Scientists Aug. 1999* Nanjing, P.R.China.
7. Piegl, L A and Tiller, W 'Geometry-based triangulation of trimmed NURBS surfaces' *Computer-Aided Design* Vol 30 No 1 (1998) pp 11-18.
8. Cho, W, Patrikalakis, N M and Peraire, J 'Approximate development of trimmed patches for surface tessellation' *Computer-Aided Design* Vol 30 No 14 (1998) pp 1077-1088.
9. Hui, K C and Li, Y 'A feature-based shape blending technique for industrial design' *Computer-Aided Design* Vol 30 No 10 (1998) pp 823-834.

10. Piegl, L A and Tiller, W *The NURBS Book* (2nd Ed.) Springer-Verlag, New York (1997).

11. Farin, G *Curves and Surfaces for Computer Aided Geometric Design* (3rd Ed.) Academic Press, San Diego (1993).

12. http://ddsdx.uthscsa.edu/dig/itdesc.html