THREE DIMENSIONAL OBJECT MODELLING FROM PLANAR SECTIONS

By

Millan Kin-Chi Yeung

Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements
for the degree of

MASTER OF SCIENCE

Department of Computer Science,
University of Manitoba,
Winnipeg, Manitoba, Canada.

ISBN 0-315-77803-2

THREE DIMENSIONAL OBJECT MODELLING

FROM PLANAR SECTIONS

BY

MILLAN KIN-CHI YEUNG

A Thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Contents

ABSTRACT

In product development, a three dimensional model is often constructed in a computer system to assist in the design or modification of the product. In the process of creating a three dimensional model to represent an existing object, several problems are usually encountered. Over-determined input data resulting from the data acquisition process cause several problems which affect data storage and processing. Modification of the object's shape may produce an unsatisfactory result because of possible discontinuities between changed and unchanged areas of the object's surface. Finally, an improper representation of the object reduces the flexibility of the design and may not be suitable for the manufacturing process in the production of the physical model of the object.

In this thesis, these problems are reduced by various methods. A heuristic data reduction scheme is introduced to reduce the impact caused by over-determination. Secondly, a set of cubic periodic splines are defined to represent the planar sections of the object and to generate the data for later surface construction. Finally, a parametric bicubic B-spline surface is created to approximate the object. Parametric bicubic B-spline surfaces provide the C^2 continuity between patches which resolves the discontinuity problem. Being locally controllable, B-spline surfaces provide flexibility to the design and modification process. Also, points on the surfaces can be interpolated with proper parameter values which makes the surface representation more adaptive to the manufacturing process.

To test these methods and procedures, a sample object was produced. The result was satisfactory and the model produced measured up to expectations. This completed the project of this thesis.

As an experiment, this thesis project illustrated the flexibility of parametric splines and explored possible applications. With further research and development, the applicability of parametric spline curves and surfaces may expand well beyond the domain

of this thesis project.

CHAPTER I
INTRODUCTION

1.1 Three dimensional object production and reproduction in manufacturing industries and biomedical applications.

In today's manufacturing industries product development is an important phase of the entire manufacturing process stream. During this phase an idea is conceptualized and the design of the product begins. Among many different design techniques, computer aided design (CAD) plays an important role in increasing the efficiency of the design process. It allows the designer to precisely construct a three dimensional object model in a computer system and to store the geometry of the object electronically. Interference between geometries can be automatically detected and corrected by a CAD system. Three dimensional views of the object can be visualized even before actually being produced. Modifications and improvements to the design can be achieved by simply changing the geometries of the CAD model. When the design is visually and theoretically satisfactory, the computer data of the geometries of the object is transferred to a computer aided manufacturing (CAM) system to produce the physical prototype. This prototype is put into the actual application for testing and analysis. If the test results are satisfactory the object design is finalized and put into production, otherwise the prototype object is thoroughly inspected and the inspected data will be used to redesign the object for the development and production of a new prototype.

There are many different techniques to reproduce a three dimensional object; in general the process is divided into three successive stages. The shape and configuration of the object is captured in the "data acquisition" stage. In the "design and drafting" stage, this captured shape and configuration is converted into either drawings and/or CAD models. Finally, the "manufacturing" stage concludes the process by production of the object with appropriate manufacturing processes.

The importance of object reproduction is increasing for various biomedical applications. Interfaced with modern data acquisition equipment such as computer tomography (CT) scanners and nuclear magnetic resonance (NMR) imaging equipment, three dimensional models of objects inside the human body such as bones, veins, organs, etc. can be reproduced for different applications in diagnostic, academic and many other areas. The object reproduction process for biomedical applications is quite similar to that for the manufacturing industries, except that the data acquisition techniques may be different as most of the objects to be reproduced are usually hidden underneath layers of skin and muscle while objects to be reproduced in manufacturing industries are usually exposed and can be physically inspected.

1.2 Data acquisition and representation of three dimensional object.

With the advancement of electronic and computer technologies, new methods to capture the shape and configuration of a three dimensional object have emerged. New advanced measuring and inspection equipment such as the computerized numerical controlled (CNC) coordinate measuring machines (CMM)[1], laser scanners[2] and stereo-vision scanners[3] are being employed to capture the information of the complex shape of the three dimensional object in the manufacturing industries. In the biomedical fields, CT scanners and NMR scanners are gradually replacing the traditional X-ray planetary photography to obtain the information regarding the internal condition of the human body. This captured information is processed and converted into the appropriate representation for further CAD and CAM processes.

Three dimensional objects are commonly represented mathematically as surfaces and this seems to be the most effective representation for computer aided operations. There are many different methods to construct surfaces in a computer system. In the following chapter several approaches are presented and their advantages and disadvantages are discussed.

1.3 Problems and requirements in object design and modification with acquired data.

In product or part design, the shape of the object is often one of the most important factors in the performance of the finished product, or in sales in the market place. For examples, the smooth aerodynamic shape of turbine blades in a turbine engine improves the efficiency of the engine and an attractive looking product may increase sales as it appeals to the customer visually. Hence the shape and configuration of the object should be able to fulfil their intended functions while still maintaining an attractive appearance. With all these constraints, the representation of the object in the CAD system must be sufficiently flexible to allow ease of modification and change.

The importance of the shape and the appearance of the product or part is well known to the developer but it is often hard to achieve because of many difficulties in the design phase of the product development process. Inappropriate representation of the object hinders the modification of the shape of the object and makes it almost impossible to change. On the other hand, designers often neglect the aesthetic appearance of the object while they concentrate on the internal design. A better representation of the object making it more amenable to change in a manner which preserves both efficiency of operation of the finished product and its aesthetic appearance is needed.

Another problem often encountered in the design of three dimensional objects is that of over-determination by measurement. This problem occurs frequently in the data acquisition stage of object reproduction due to the nature of the operation of the data acquisition equipment. Processing data which over-determines an object poses several problems in the CAD model construction process as well in the CAM process to produce the object. A sensible method to eliminate or reduce the severity of problems caused by such an over-determined data set is required.

1.4 A proposed solution with planar contours and parametric surfaces.

In this thesis, a compromise solution is proposed to these problems by integrating the concepts of data-point reduction, closed planar contours and parametric surfaces to construct an approximate representation of a three dimensional object. This approach is applicable to the design for both object production and reproduction. Several tasks are designated to implement this solution.

The first task is specifically designed for the reproduction of an object. It reduces the over-determined data for each set of co-planar points which is obtained through data acquisition. The data-reduction scheme used in this stage is referred to as "Tolerance dependent data point elimination[4]". This scheme preserves some of the original data points of the contour and allows the user to control the tolerance to eliminate the unnecessary data points. With an appropriate tolerance, the number of data points can be reduced to a manageable quantity.

The second task is to fit a closed parametric cubic spline curve to each set of data. The corresponding angle measurement of the data is used as the parameter to parametrize the curve. In this formulation, a point on the curve can be interpolated by an input angle with the origin located inside the closed curve. These curves are used to generate a mesh of grid points for the construction of the surface.

The third task is to apply interpolation on the mesh for surface creation. Since the closed cubic splines created in the second task are generated after the data reduction process, the knots of the splines are no longer aligned to each other. It is then necessary to generate a new set of properly aligned points. This can be done with interpolation techniques. With all the corresponding points aligned across all the splines, a well ordered mesh of grid points is generated.

The final task is to fit a surface to this mesh of points which approximates the external surface of the object. In order to create a smooth and flexible surface, a bicubic parametric surface is used. This surface has only two edges on both top and bottom which are formed by the first and last closed planar curves. Both ends of the object are usually constructed with some relatively simple connective features to interface with other objects or presented as flat surfaces, allowing the user some freedom to tailor the end surfaces.

This approach of using heuristic data-reduction, closed planar contours and parametric surfaces addresses the problems induced by over-determined data sets and also allows the user to make changes to the object easily while still maintaining a nice and smooth appearance. The implementation of this method is relatively simple and thus could be easily integrated into most of the CAD/CAM systems.

CHAPTER II
PROBLEMS IN THREE DIMENSIONAL OBJECT REPRESENTATION AND EXISTING SOLUTIONS

2.1 Data acquisition methods and the problems they induce.

Many different methods and computerized equipment such as laser scanners, stereo-vision scanners, CMMs, CT scanners and NMRs are used to capture data from a three dimensional object. Laser or stereo-vision scanners and CMMs output points in three dimensional space to represent the surface of the object whereas the output of CT scanners and NMRs is a series of two dimensional images which represent various cross sections of the object. Although such equipment is very efficient in the data capturing process, its output usually over-determines the measured object.

There are two types of laser or stereo-vision scanners, the two axes line scanner and the three axes mesh scanner. Depending on the type of scanner used, points approximating the geometry of the object are generally produced on a straight line, i.e. the projection onto the x-y plane of the sampled profile is a straight line segment, or on a uniform rectangular grid, i.e. the domain on the x-y plane of the sampled surface patch is a uniform mesh. Points on a straight line are sampled with a two axes scanner[2], which is stationary, with its sensor directed to the object. It makes a single scan pass across the object surface to obtain an array of points containing the (x,y) coordinates and elevation of points on the surface. This array of points approximates the outer profile of the object's cross section along the scanning path. The entire surface of the object can be approximated by incremental movement of the scanner in an orbit around the object, making a new scan pass at each increment.

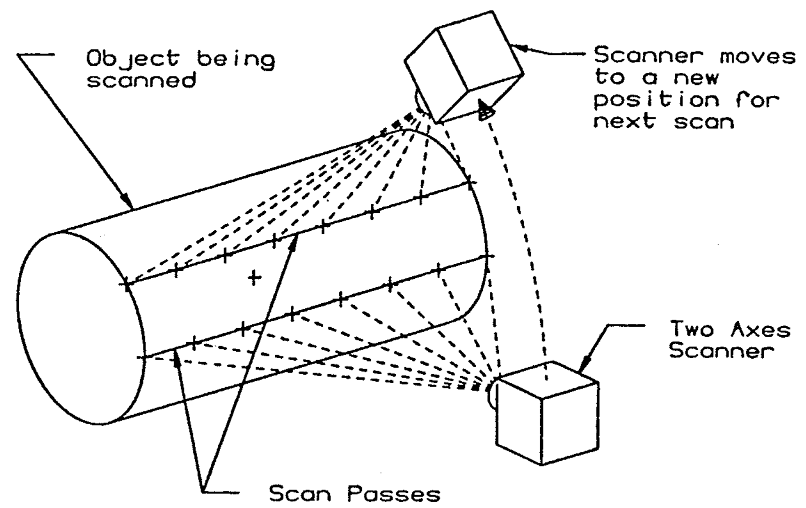


Figure 2.0

Figure 2.0 shows an example of a cylindrical type surface being approximated. With a three axes scanner[5], sensors scan along multiple paths in two directions that are orthogonal to each other just like the X and Y axes in the Cartesian coordinate system. These passes cover a surface patch of the object on a rectangular domain and produce a set of surface elevations which correspond to points on a uniform rectangular grid. The entire affected area is approximated in one scanning process and it is not necessary to move the scanner. Figure 2.1 illustrates an example of this scanning process.

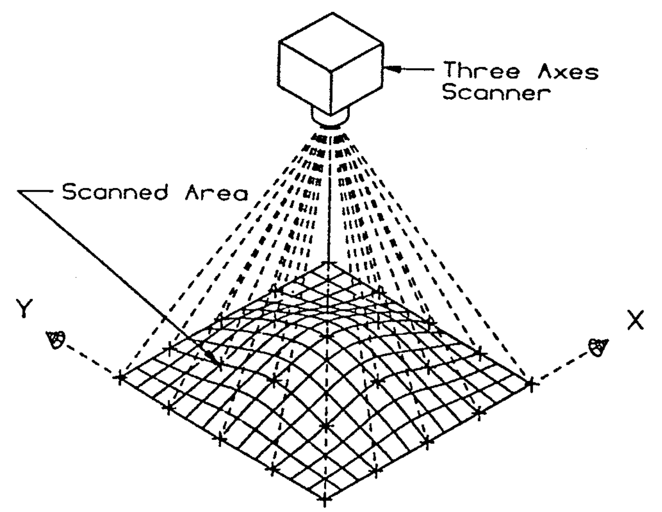


Figure 2.1

The output of either methods is a well ordered set of grid points. For some scanners, the distance between points is controllable but constant for each scan. This feature limits the number of output points per scan. In most cases, the distance between points is kept to a minimum in order to capture as much information as possible of the object's surface. The problem is thus over-determined. An over-determined data set occurs when the object can be well represented by fewer than the given number of data points. Figure 2.2 is an example of an over-determined representation generated by a scanner. A line can be approximated by two points but the scanner may produce ten points along the line because of the set distance between points.

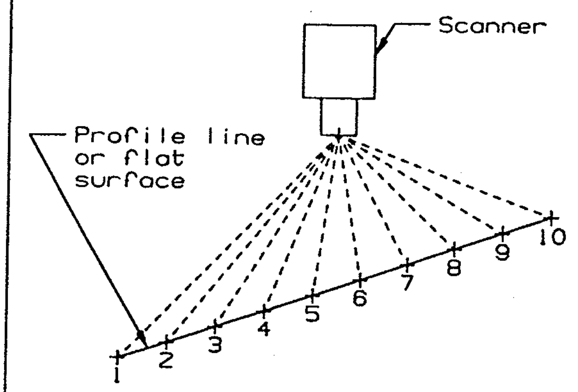


Figure 2.2

The operational concept of computerized numerical controlled CMMs is quite similar to the laser or stereo-vision scanners. Instead of using the laser beam or stereo-video cameras to scan the object, it uses a mechanical touch sensor probe to digitize the object. Hence the over-determined problem exists.

CT scanners and NMRs are used mostly in medical applications to observe the internal condition of the human body. A series of two dimensional raster images are produced for each different cross section of the object scanned. A three dimensional model of the object can be created by combining these images with the correct alignment and with the same distance between sections. The process is divided into three phases [6]. During the first phase the outline of each cross section is determined by image analysis. The second phase aligns these outlines into a three dimensional model such that all the cross sections of the object are represented and placed in the corresponding positions. The final phase fits a surface onto these outlines to complete the three dimensional model of the object. The problem of over-determined data sets usually occurs in the image analysis phase. During this phase an experienced operator or medical doctor examines each image and identifies the outline of the object. The pixels which constitute this outline are then converted into data points to represent the cross section. Since the observed pixels are very dense, the conversion generates a massive number of points for the representation and therefore the over-determined problem occurs.

The over-determined data set causes problems in CAD model building as well as in the CAM process. In the CAD model building process these data points are used to fit a closed curve to represent the cross section of the object. This massive data set floods the processor and increases the processing time significantly. Performance is decreased and data storage is wasted.

In the CAM process, these excessive points introduce problems to the CNC machining process when they are used to generate the tool path (or the NC program). In most of the CNC controllers, cutting motions are restricted to be either linear or circular. In general, a series of linear motions is generated to interpolate the non-linear curves. When using the scanned points directly to generate the NC program, each consecutive point becomes the end point of a linear path emanating from the current position. Hence the excessive points will generate excessive motions for the CNC machine. These excessive move commands inflate the size of the NC program which could exceed the storage capacity of the CNC controller and thus the entire NC program would not be stored in the controller at one time. As the distance between points is minimized in order to capture more information of the object's surface, the distance of the linear motion for each NC command generated is also shortened. This causes the data buffer in the CNC controller to underflow. This buffer underflow problem occurs when the machine completes the physical motion of the current command and is ready to execute the next one but the next command has not yet been completely transmitted. The current cutting movement stops and the tool waits for the next command before making the next move. This stop-and-go motion causes oscillation along the tool path and affects the quality of the finished surface. Intense oscillation may actually damage the machine tool.

2.2 Data point reduction with heuristic approaches.

One approach to eliminate or to reduce the severity of these problems caused by the over-determined data set is to eliminate the excessive points before converting the data set into a tool path. There are different methods to reduce the number of data points but the integrity of the data set must be maintained; the desired original data points should not be altered or destroyed and the loss of surface information should be minimized. In the past decades, many methods to deal with this type of data reduction were developed. A good collection of these methods were published by McMaster[7]. Descriptions of some of these methods now follow.

The Strip Algorithm by K. Reumann and A.P.M. Witkam[8] starts by forming a line connecting the first two points in the data point array and refers to it as the critical line for the starting point. In Figure 2.3 the critical line is formed by joining P_a and P_{a+1}.

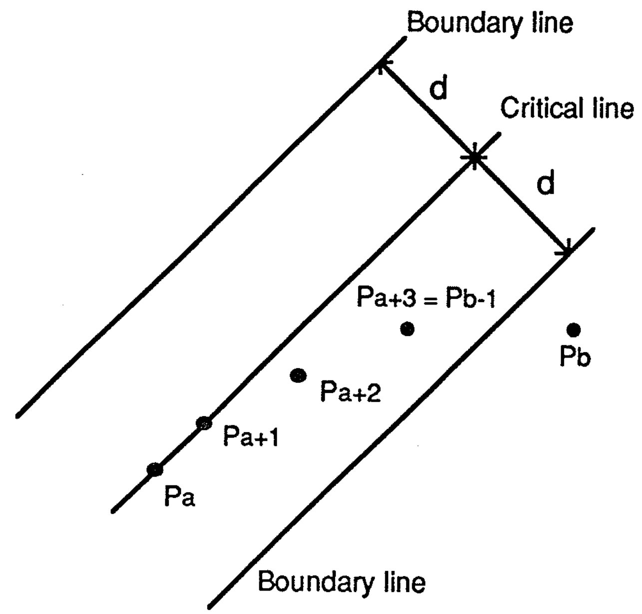


Figure 2.3 Strip Algorithm

Two boundary lines are drawn parallel to and on either side of the critical line with the distance d (where d is the tolerance). Beginning with point P_{a+2}, the remaining points are examined until a point P_b ($a+2 \le b \le n$; n is the number of data points) is found, where P_b is the first point lying outside the region formed by the boundary lines or P_b is the final point, P_n, in the array. All points within the region except the points P_a and P_{b-1} are eliminated. The point P_{b-1} is then set to P_a and the process repeated until $P_b = P_n$.

Williams' algorithm[9] attempts to reduce the input data point array based on the following criteria :

(1) $Q_0 = P_0$ and $Q_m = P_n$, where Q is the new array of remaining points after data point reduction and P is the original array of points, n = size of P and m = size of Q.

(2) For any points Q_i and Q_{i+1} ($0 \le i < m$), where $Q_i = P_a$ ($0 \le a < n$) and Q_{i+1}

$= P_{b-1}$ $(a < b-1 \leq n)$, all the points P_c $(a < c < b-1)$ lie within distance d (where d is the tolerance) of the line segment formed by points Q_i and Q_{i+1}, or equivalently, within distance d of the line segment formed by points P_a and P_{b-1}.

Figure 2.4 shows the concept of this algorithm and the result of a sample reduction. The points P_{a+1} to P_{a+6} are eliminated (note that P_{a+4} is not labelled) as they are within the 2d tolerance as shown. The remaining points are P_a and P_{a+7} which corresponds to the points Q_i and Q_{i+1} of the new array after the data point reduction.

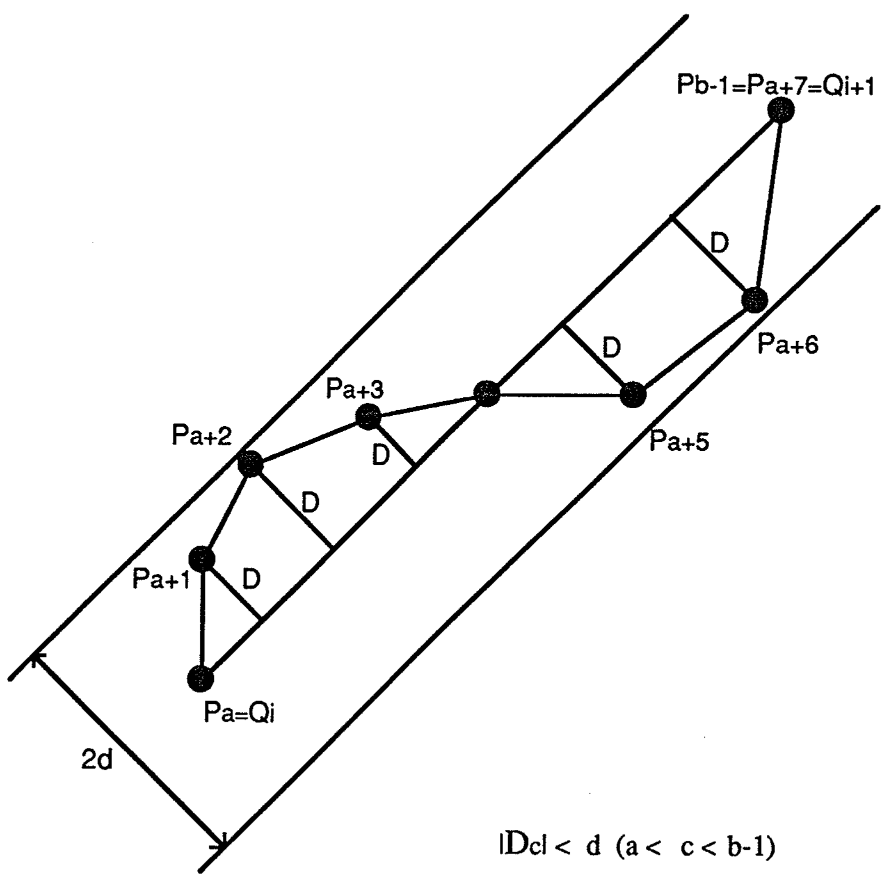


Figure 2.4 Williams' Algorithm

To implement this concept, Williams uses the method illustrated in Figure 2.5. First draw a circle centered at P_{a+1} with radius d and then draw two lines through P_a, tangent to this circle, which conforms to an angular displacement of L_1. Similarly, draw a circle at P_{a+2} and two lines through P_a tangent to the circle. This conforms to another angular displacement, L_1', which intersects L_1. The intersection is labelled as L_2. Repeat this procedure for P_{a+3}, P_{a+4}, ... and so on until P_k such that L_k is empty. All intermediate points between P_a and P_{k-1} are eliminated.

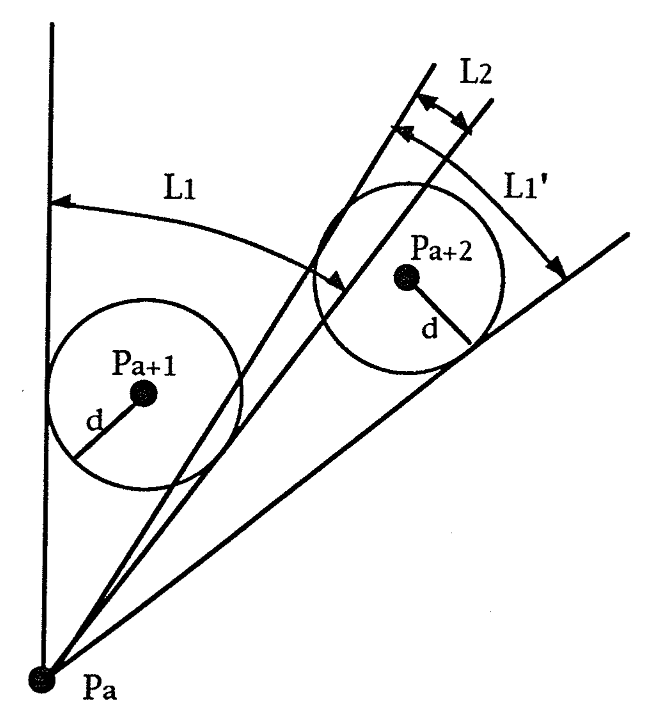


Figure 2.5 Definition of L2

The Fan Data-Compression[10] is similar to Williams' algorithm. Both use the tolerance region method as shown in Figure 2.5. Additionally, the Fan Data-Compression algorithm reduces the region L_1 by projecting it like a fan from point P_a to an area that is beyond the circle centered at P_{a+1}.

As illustrated in Figure 2.6, a point P_i is rejected if it is neither the first nor the last point, and if it is within the tolerance d to the line segment joining the point P_{i+1} and the last point retained which was labelled as P_0. Otherwise, P_i is kept and is labelled as the new P_0.

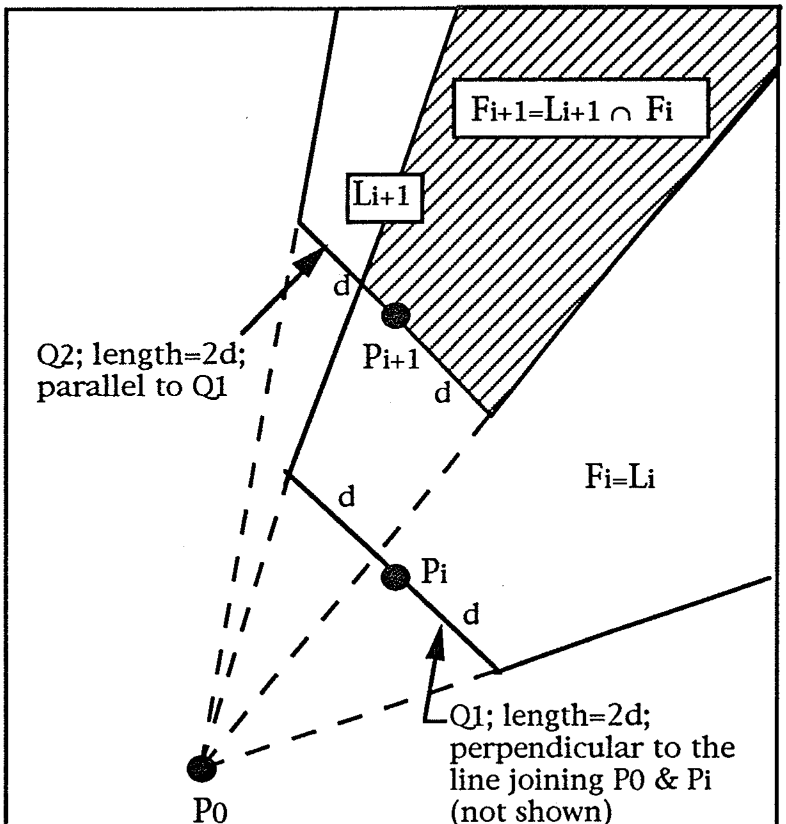
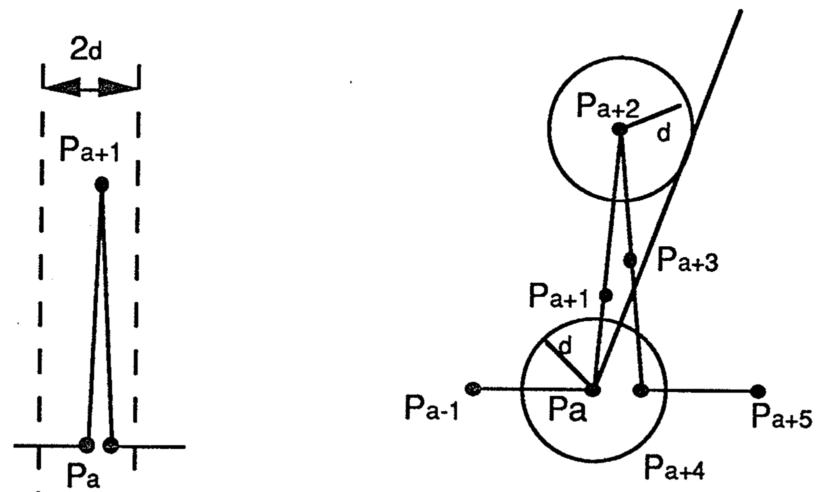


Figure 2.6 Fan Data-Compression Point Elimination Criterion

To determine if P_i can be eliminated, a feasible region F_i is constructed such that if P_{i+1} lies in F_i, then P_i is eliminated. To construct F_i, the procedure first constructs a local region L_i such that every line segment connecting P_0 to a point in L_i passes within d of P_i. This is achieved by drawing a line segment Q_1, of length 2d, through P_i, and perpendicular to the line passing through P_0 and P_i. Rays from P_0 which touch the end points of Q_1 are then drawn. The partially infinite region bounded by these two rays and Q_1 defines L_i. If P_i lies within d of P_0 then L_i is taken to be the entire plane. F_i is formed by taking the intersection of L_i and F_{i-1}. Note that F_0 is the entire plane. Next is to construct F_{i+1}, a line segment Q_2, through P_{i+1}, of length 2d and parallel to Q_1 is defined. Define L_{i+1} as before, then the intersection of L_{i+1} and F_i defines F_{i+1}. Repeat this procedure to examine P_{i+2}, P_{i+3}, ... and to define F_{i+2}, F_{i+3}, ... until F_k is empty, then the last point P_k is retained



Strip Algorithm Williams' Algorithm

Figure 2.7 Inflection Point Problem

These algorithms work in most cases but suffer two major inadequacies. The Strip algorithm and Williams' algorithm can not handle certain inflection point problems as illustrated in Figure 2.7. P_{a+1} is wrongfully eliminated with the Strip algorithm while Williams' algorithm eliminates P_{a+1}, P_{a+2} and P_{a+3} but P_{a+2} is supposed to be kept. Finally, the methods used in these algorithms to determine which points are to be eliminated are quite complex. Hence the implementation of these algorithms may be difficult. In order to overcome these problems, a simple but effective method is used for the data point reduction of this project and it is described in the later section of 3.2.

2.3 Shape modification and surface representation of objects.

As mentioned in chapter one, the representation and the shape of the object are very important to the success of the product in either application or in marketing. In object design, generally the original shape of the object designed is attractive and smooth since it is a product of the optimism and desire of perfection of the designer. As the development progresses, the shape of the object usually requires changes and modifications for practical reasons. For instance, some functions of the object can not be performed because of the weakness of certain areas of the object. Strengthening the object by increasing the thickness of the affected areas may restore the functionality of the object but also changes the shape of the object in those areas. Another practical reason is that a component may not fit into the object because of the inadequate space allowed in the original design. Enlarging the internal cavities of the object to accommodate components means that the external shape of the object is also expanded. All these changes require the modification of the shape of the object. To allow such modifications, the surface representing the object should permit the designer to contract or expand the object partially or completely.

The change of the entire surface is not too difficult. This can be accomplished by scaling the entire data set representing the surface of the object. The entire surface is affected or controlled by the operation. Modification to certain areas of the surface without disturbing other areas requires an entirely different solution. This type of modification requires the control over the local area only. This partial control is referred to as local control. When the change is made, only the shape of a small area is modified while the shape of the rest of the object remain unchanged. This local control ability is critical to the entire design of the object as it allows the modification to be made to a particular area of the surface without distorting the already satisfactory designed areas. The implementation of functions with local control ability is not trivial; the following constraint must be considered.

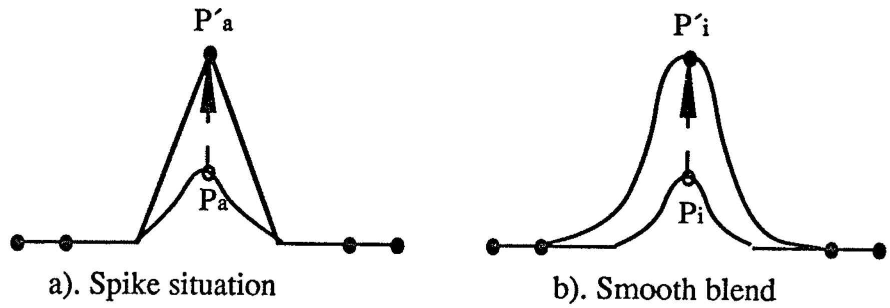


Figure 2.8 Local Control

When making changes to certain areas without disturbing other areas, the effects on the shape of the surface should be considered. As shown in Figure 2.8, two situations are compared. In a) the point P_a is pulled up to P'_a. The surface follows the movement of P_a and stretches to P'_a but the base of the affected section does not change. This causes a spike condition as shown. The modification is local since the most areas are not disturbed but the shape of the surface becomes undesirable. The change made in b) is more acceptable. In this case, when the point P_i is pulled up to P'_i, the surface is stretched upward and the curvature changes gradually. The base of the changed section blends into the unchanged areas smoothly and thus the shape of the surface remain smooth and visually appealing. To achieve a smooth blend, the designer should either patch the base of the changed area with some small fillet surface patches or uses a surface representation that allows local control and which would automatically “smooth-blend” the base into the unchanged area.

Representing a three dimensional object in a computer system is not trivial; surfaces are usually used. Two types of surface patch representations commonly used in computer graphics systems are triangular tiles and parametric surface patches.

In the triangular tile approach a surface is described by many appropriate sized triangles. Depending on the distribution of the data points acquired, the number of triangles and their sizes vary from one area to another. A triangle is composed of three vertices and three edges or arcs. With the provided or captured data set, every three data points form a triangular tile. Depending on the triangulation algorithm used, a single point can be the vertices of many different triangles. More triangles are generated if the concentration of data points is high and the surface is better approximated. With fewer data points, the number of triangles is proportionally decreased. This type of representation is often used for rendering the object.

The parametric surface patch approach uses a mathematical based representation of a surface. Any point on the surface patch can be interpolated mathematically. The analytic nature of parametric surface patches allows smoother representation of surfaces than the triangular patch approach described above. The most commonly used parametric patches are bicubic. They are generalizations of parametric cubic curves. Hence the definition [11] of the parametric bicubic surface patch is also derived from the definition of the parametric cubic curve. A parametric cubic curve has the form :

$$Q(t) = T \bullet M \bullet G$$

where Q is the cubic curve; T is the parametric vector $[\ t^3\ t^2\ t\ 1\ ]$; M is a 4x4 basis matrix; G is a four element vector of geometric constraints and is called the geometry vector. In this case, G is a constant. To define the parametric bicubic surface patch, replace the parameter t with s, hence $Q(s) = S \bullet M \bullet G$. Now we allow the points in G to vary along a three dimensional cubic path parametrized with respect to t, thus :

$$Q(s, t) = S \cdot M \cdot G(t) = S \cdot M \cdot [\, G_1(t)\ G_2(t)\ G_3(t)\ G_4(t)\,]^T \tag{2.1}$$

For a fixed t, Q(s, t) is a curve because G(t) is constant. When t takes on some new value, Q(s, t) is a different curve. Repeating this for arbitrarily many values of t between 0 and 1, an entire family of curves is defined, each arbitrarily close to another curve. The set of all such curves defines a parametric bicubic surface. Now each $G_i(t)$ can be represented as

$$G_i(t) = T \cdot M \cdot G_i \tag{2.2}$$

where $G_i = [\, g_{i1}\ g_{i2}\ g_{i3}\ g_{i4}\,]^T$ and g_{ij} is the jth element of the geometry vector for curve $G_i(t)$. Transposing the Equation (2.2) $G_i(t) = T \cdot M \cdot G_i$ and substitution into Equation (2.1) leads to :

$$Q(s, t) = S \cdot M \cdot G \cdot M^T \cdot T^T \;\; ; 0 \leq s, t \leq 1.$$

where G is a 4x4 matrix with elements $[\, g_{ij}]$ for $1 \leq i \leq 4$, $1 \leq j \leq 4$.

This is the general definition of a parametric bicubic surface. Different geometry vector G and basis matrix M may be chosen to obtain a particular surface patch, e.g. Hermite, Bézier or B-spline. The bicubic surface patches as used in this thesis will be discussed more fully in section 4.5.

2.4 Existing methods of surface construction.

There are several methods of constructing three dimensional surfaces. The data set acquired for the construction of the surface in the context of this work is restricted to a set of closed planar contours. Hence only methods which construct surfaces based on planar contours are discussed.

The major problem of constructing the surface between two adjacent planar contours is matching corresponding points between them. The numbers of points on both contours may not be the same, and the orientation or position of these points may not be aligned to each other. In general, the problem may be stated as follow :

Let one contour be C_P defined by the sequence of m distinct contour points P_0, P_1, ..., P_{m-1}, and let contour C_Q be defined by the sequence of n distinct contour points Q_0, Q_1, ..., Q_{n-1}. Note that P_0 follows P_{m-1} and Q_0 follows Q_{n-1}, so the indices of P are modulo m and indices of Q are modulo n. For each point P_i, there are n possible correspondences to points Q_0, Q_1, ..., Q_{n-1} and similarly for each point Q_j there are m possible correspondences to points P_0, P_1, ..., P_{m-1}. This introduces a problem of defining appropriate correspondences.

The first solution presented is a generalized method of triangulating the surface with planar contours which was developed by H. Fuchs, et al [12]. It constructs the surfaces using triangular tiles to connect two adjacent planar contours. This method is repeated for all the contours and the entire surface of the object is constructed.

The process starts with orienting the tiles. The notation $\{P_i, P_{i+m1}, Q_j\}$ is used to represent the tile consisting of the contour segment connecting P_i and P_{i+m1} (where i+m1 means i+1 modulo of m), the left span P_iQ_j and the right span $P_{i+m1}Q_j$. Similarly, $\{Q_j, Q_{j+n1}, P_i\}$ represents the tile consisting of the contour segment connecting Q_j and Q_{j+n1}, the left span $Q_j P_i$ and the right span $Q_{j+n1}P_i$. To construct the surface with these tiles, they must satisfy the following two conditions :

(1) Each contour segment appears in exactly one tile in the set.

(2) If a span appears as a left (right) span of some tile in the set, then it also has to appear as a right (left) span of at least one tile in the set.

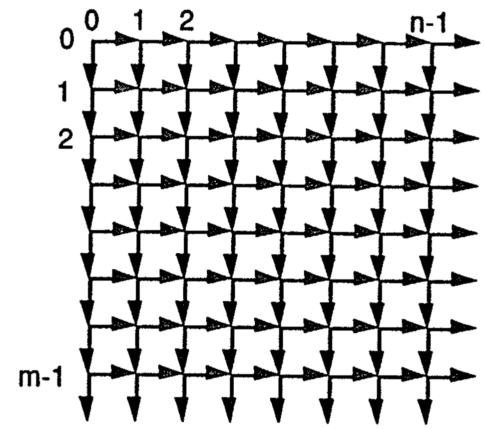


Figure 2.9 Toroidal graph

To define the tiles that satisfy these two conditions, this method uses a Toroidal graph to solve the tile selection problem. Define a directed graph G<V, A>, in which the vertices correspond to the set of all possible spans between the points $P_0, P_1, \ldots, P_{m-1}$ and the points $Q_0, Q_1, \ldots, Q_{n-1}$, and the arcs correspond to the set of all the possible tiles.

$V = \{ V_{ij} \mid i = 0, 1, \ldots, m-1; j = 0, 1, \ldots, n-1 \};$

$A = \{ <V_{kl}, V_{st}> \mid \text{either } s = k \text{ and } t = l +_n 1 \text{ or } s = k +_m 1 \text{ and } t = l \}$

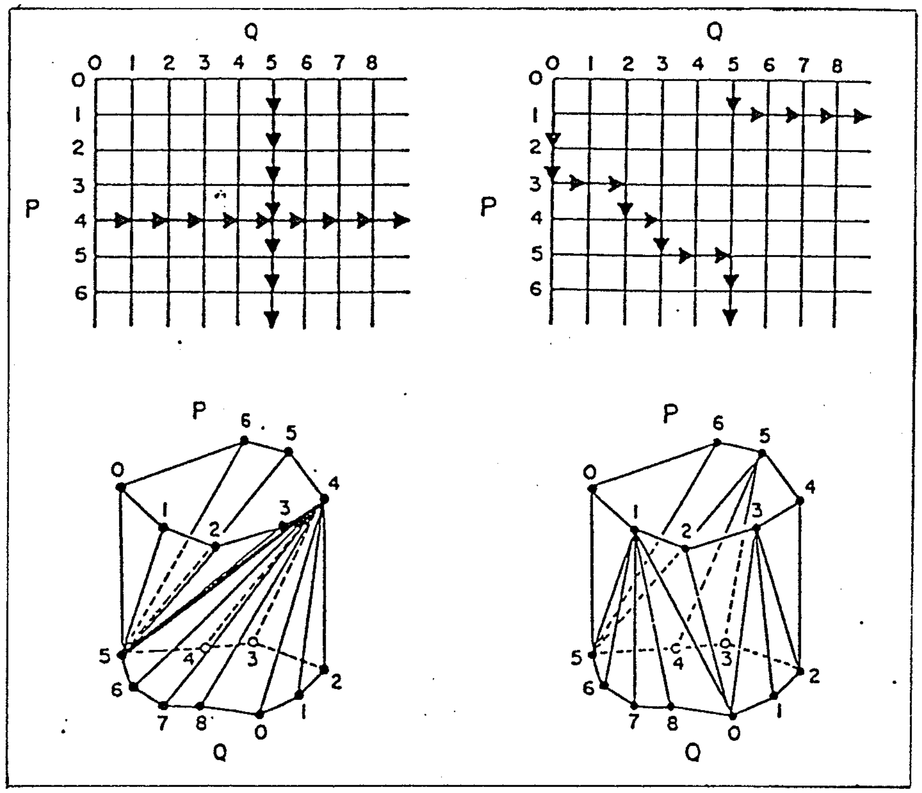
where V_{ij} corresponds to the span P_iQ_j and $<V_{kl}, V_{st}>$ corresponds to the tile with the left span P_kQ_l and the right span P_sQ_t. Figure 2.9 shows the graph G in a grid form in which the vertical arcs beyond row m-1 point back to row 0 and the horizontal arcs after column n-1 point back to column 0. This condition is not shown in the drawing. In the graph, V_{ij} is referred to as the vertex at row i and column j; similarly the arc $<V_{ij}, V_{i+_m1,j}>$ is a vertical arc between the rows i and $i+_m1$, and the arc $<V_{ij}, V_{i,j+_n1}>$ is a horizontal arc between the columns j and $j+_n1$.

The acceptable surface between these two planar contours is formed by the set of tiles S that satisfies :

(1) S contains exactly one horizontal arc between any two adjacent columns and exactly one vertical arc between any two adjacent rows.

(2) S is eulerian; That is S can be traversed by a closed walk in which every arc of the graph occurs exactly once.

Figure 2.10 shows two possible solutions in the graph. Both surfaces are valid and are acceptable since they satisfy the two conditions but solution a) may not be desirable because it actually forms two cone like surfaces. There may be many acceptable surfaces, so additional criteria should be used to choose an optimal or near optimal one. The additional constraint added is a cost C to associate with each arc $<V_{kl}, V_{st}>$ of the graph G as $C(<V_{kl}, V_{st}>)$.



a) b)

Figure 2.10 Acceptable solutions and surfaces.

The cost of an acceptable surface or an eulerian trail will be the sum of the costs of the arcs traversed by it. A surface is deemed the most desirable if it has the minimum cost. This concludes this generalized solution.

Some optimal solutions which are based on this generalized method were suggested in an attempt to perfect the surfaces generation with planar contours. These solutions follow the Fuchs' method to define acceptable surfaces but uses the specific scheme of a cost function to select the desirable one.

H. Fuchs himself selects the optimal surface such that the cost function C is minimal. The path in the graph that generates the optimal surface based on this selection scheme is referred to as the minimal weighted path. He uses the surface area to define C and therefore the optimal surface will have the minimum area among all the acceptable surfaces.

Alternatively, E. Keppel[13] proposed that the optimal surface be an acceptable surface such that the cost function C is maximal. This approach is referred to as the maximal weighted path. He uses the volume bounded by the two planes of the planar contours and the acceptable surface to define the cost function. The acceptable surface that enclosed the largest volume is chosen to be the optimal surface.

Finally, the Christianson Heuristic[14] suggested by H. Christianson and T.W. Sederberg has been known as the best for contour pairs which are coherent in size and shape, and mutually centered. The cost function is defined based on the length of the span, and the tile with the shorter span is selected. In this way, the cost function is locally minimized at each vertex and thus improves the processing efficiency.

All these suggested solutions use triangular tile patches to construct the surface. This approach is particularly convenient for rendering the surface to display the object. With sufficient memory space in the computer, it can use all the original data points to maximize the details of the object. Since there is no physical process involved other than computer processing, the construction of the surface can be achieved relatively quickly. Unfortunately, this representation may not be suitable for the CAM process to produce the physical model of the object because it does not provide a convenient mathematical tool path description as required for computerized manufacturing equipment. The huge data volume that maximizes the effect of this representation will have undesirable effects during the manufacturing process as mentioned in the problem of over-determined data sets. It will also slow down the manufacturing time considerably as more patches are created and more physical machining motions will be generated. The physical motion on the machine is much slower than the computer process, therefore the surface construction process may seem fast on the computer but it will take a much longer in producing the physical model of the object. Finally, the rigidity of the triangular tile dictates a fixed pattern of the cutting path to drive the manufacturing equipment which is not desirable to the manufacturing process.

2.5 Summary.

Data acquisition with advanced equipment is fast and accurate but usually causes a problem of over-determined representation. Laser scanner, stereo-vision scanner and CMM types of data capturing equipment produce a grid of points from the shape of the object. The distance between points is usually minimized in order to capture the maximal amount of information. CT scanners and NMR equipment capture information of the internal structure of the object and output a series of 2D images of various cross sections of the object. The data points which constitute the outline of a cross section is defined by a chain of appropriate pixels selected from the images. An excessive number of points usually results from the process because of the tightly clustered pixels. Hence the problem of an over-determined data set occurs. Subsequently the over-determined data set introduces many problems to the CAD system in the object design process as well as to CNC equipment in the CAM process.

Reduction of excessive data points is an acceptable solution to this problem. Several data reduction algorithms were considered. The Strip algorithm developed by K. Reumann and A.P.M. Witkam uses boundary lines with a given tolerance to form a strip which contain the unwanted points for later elimination. The Williams' algorithm uses projected angular displacements to eliminate the unwanted points. Both algorithms work in most cases but can not handle the inflection point problem. The Fan Data-Compression algorithm uses feasible regions to determine the rejection of data points. It's principal concept is similar to Williams' algorithm. Despite the simple concepts of these algorithms, they are quite complicated to implement.

The shape of an object is important to application and to marketing. Modification of a portion of the object's surface may cause distortion of the entire surface. To maintain the desired shape of the unaffected area when making partial changes to the surface, a flexible surface representation with local control ability is required. Triangular tile surface patches and parametric bicubic surface patches are popular types of surface representations used in most computer graphics systems. With triangular tile surface patches the entire surface is composed of many triangles while with parametric bicubic surface patches the surface is defined mathematically in terms of cubic functions.

The construction of a surface from planar contours is not a simple task. The generalized method of creating triangular tile patches with planar contours by H. Fuchs Z.M. Kedem and S.P. Uselton applies a toroidal graph as well as a cost function to define the acceptable surfaces between two contours. Fuchs uses the minimal weighted path approach with the surface area as the cost function for the selection of the optimal surface while E. Keppel proposed the maximal weighted path approach; his cost function is defined by the volume bounded by the contours and the surface. The Christianson Heuristic uses the length of the span to define the cost function which improves the processing time.

All these surface construction algorithms generate a triangular tile surface which is efficient for rendering the object in order to display the surface but the triangular tile surface is not suitable for CAM processing. A more flexible surface representation such as a parametric bicubic surface is more appropriate for the problem considered in this thieis.

CHAPTER III

THREE DIMENSIONAL OBJECT REPRESENTATION WITH PLANAR CONTOURS AND PARAMETRIC SURFACES.

3.1 Introduction.

In search of a data representation for a three dimensional object which is suitable for CAD as well as CAM processes, the triangular tile surface patch was considered but does not seem to be applicable to most of the CAD/CAM operations, such as NC tool path generation. NC tool path generation is based on the geometry of the surface. One must generate a sequence of motion commands to drive the CNC machine in the production of the object. This sequence of motion commands is defined by the corresponding points on the surface. The locations of these points vary from one tool path to another even though these tool paths may drive the same machine or cut the same surface. The number of points required to form the tool path depends on the tolerance of the cutting accuracy and the patterns of the path. The normal vector to the surface at every point may also be required to interpolate the tool axis orientation of the cutter depending on the type of machine and tool used. There are many other parameters and factors in the tool path generation that can affect the number of points, the locations of the points and the pattern of the points required. Hence, in general, these points are defined dynamically during the tool path generation process. The triangular tile surface constructed with the original data points will be inadequate for these requirements. A more flexible type of surface is desirable.

The features of parametric bicubic surfaces, particularly B-spline surfaces, make them suitable for the representation or construction of three dimensional objects which in turn can be used for the CAD and CAM processes. A cubic B-spline surface is a parametric bicubic surface which has C^2 continuity. The entire surface of the object is represented by a number of B-spline surface patches and the C^2 continuity assures a smooth blend between patches. The structure of a B-spline surface provides some degree of local control such that when the location of a control vertex changes only the neighboring patches are modified. For the generation of a tool path in the CAM process, any point on the patch can be interpolated mathematically for tool point locations; its corresponding normal vector to the surface can be calculated to define the orientation of the tool axis. Hence in this work, a B-spline surface will be used to represent the three dimensional object to be built.

As discussed in the previous chapters, the data acquired for the construction of a three dimensional object is a set of points. The configuration of this set of points is a well ordered grid with rows and columns and the coordination of these points detail the shape of the object. In order to construct the surface directly, a closed planar contour is constructed for every row of points by joining them with straight line segments. A triangulation algorithm can be applied to these contours to construct a triangular tile patch surface. Unfortunately, the acquired data set is usually over-determined as discussed in the previous chapter. In order to reduce the data points while maintaining a well ordered grid pattern, a planar contour can be used to generate the new grid points for each row after the data reduction. Cubic spline segments are used for the construction of the contour rather than straight lines to allow for a smooth fit to the remaining points. The measuring devices considered are unable to measure contour with "kinks" in them as shown in Figure 3.0, therefore this condition is not being considered in the solution.

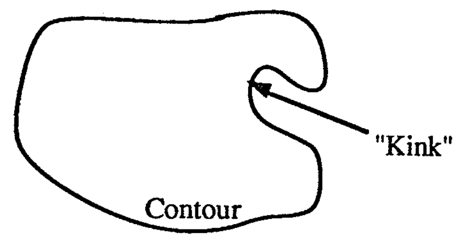


Figure 3.0
Kink condition of a contour.

However, objects produced by CAM process can usually be orientated so that the points P_i; i=0,...,n-1, on the circumference of each cross-section, satisfy the following property as illustrated in Figure 3.1.

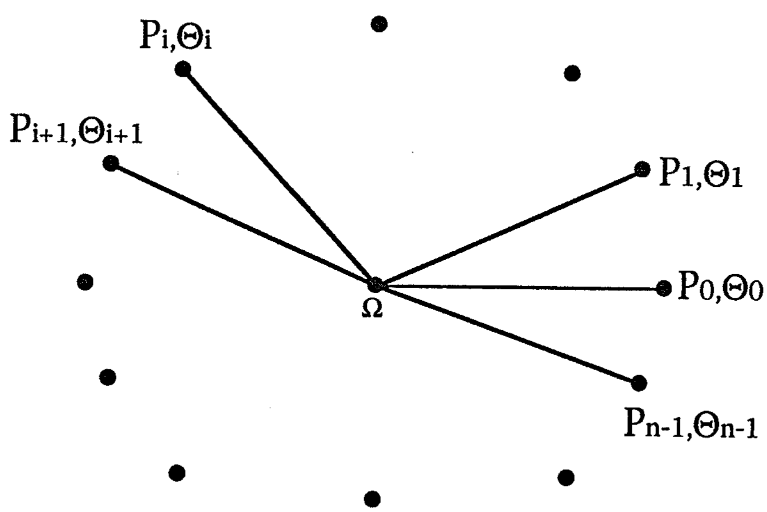


Figure 3.1

Let Ω be the centroid of the closed polygon $P_0, P_1, \ldots, P_{n-1}$ and let Θ_i be the angle measured anti-clockwise, which is formed at Ω by the line segments ΩP_0 and ΩP_i. The angles $\Theta_0, \Theta_1, \ldots, \Theta_{n-1}$ then satisfy

$$0 = \Theta_0 < \Theta_1 < \ldots < \Theta_{n-1} < 2\pi$$

These angular measurements are used as the parameters for the parametrization to interpolate new grid points. With the origin located inside all the contour lines, this method creates points on the contour curves at angles specified by the user. Specification of a uniform angular increment for all the contours leads to a well ordered grid pattern. Finally, these grid points are used as the control vertices to construct a B-spline surface. This surface which approximates the shape of the object can then be used for modification or re-design of the object and for the CAM processes. Since the model generated is only for the re-design and/or design of the object, the exact accuracy of the representation matching the existing shape of the object is irrelevant for this project. Hence the B-spline surface generated is acceptable even though it does not interpolate the input data. On the other hand, a similar approach that constructs a B-spline surface which will interpolate this new set of well ordered grid points may be achieved by defining a new set of control vertices based on the existing grid. For example, the method developed by B.A.Barsky and D.P.Greenberg[15] can be applied for this purpose. Currently a collaborative project to construct such a surface to accurately represent a three dimensional object is being developed and implemented at the National Research Council of Canada.

The flow chart on Figure 3.2 shows the processes of this project and how they will be implemented.

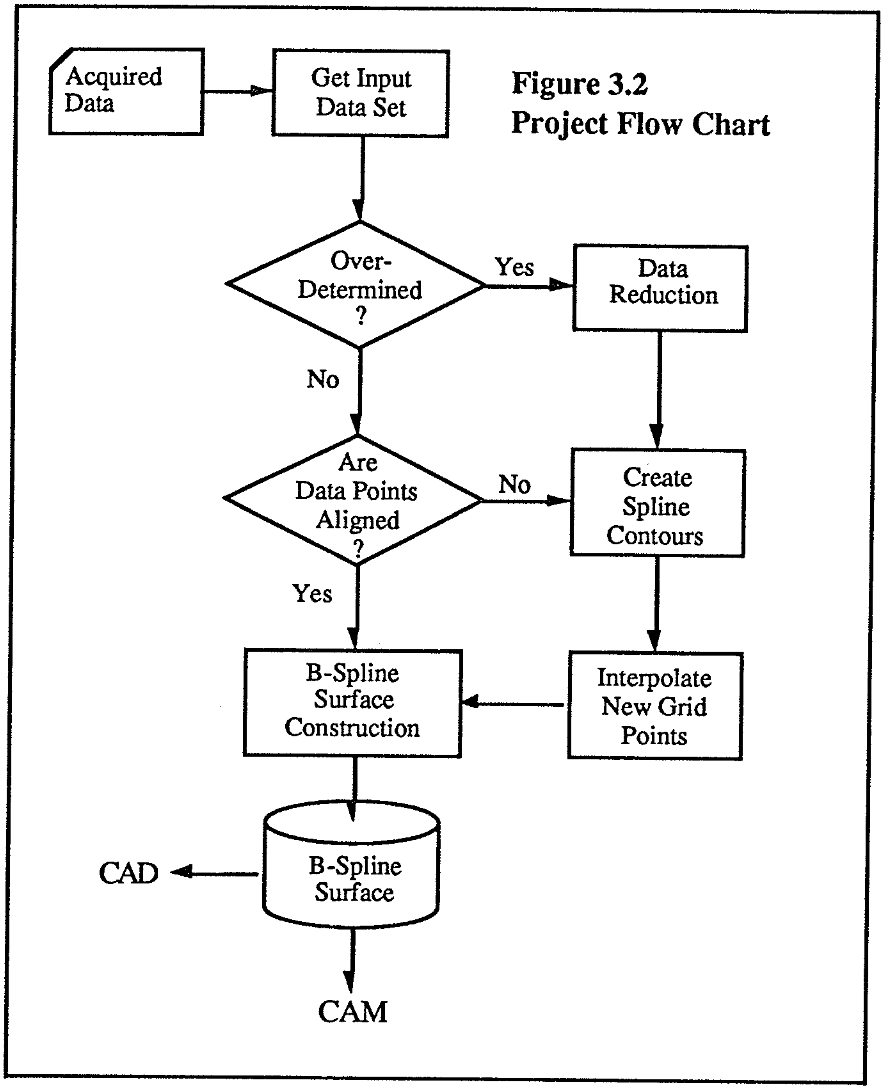


Figure 3.2
Project Flow Chart

3.2 Data point reduction of co-planar points.

In object reproduction, the data acquisition process usually tends to acquire more data points than the representation requires. Data-reduction is a process to reduce the number of points. The data-reduction algorithms discussed in chapter two do not address the inflection point problem satisfactorily and are also difficult to implement. A more suitable scheme, is the "Tolerance dependent data-point elimination" as presented by Yeung and Walton[4] which is based on the "longest line" method described by Meek and Walton[16]. It uses a given bounding tolerance to determine which points are to be eliminated along the contour. The principle concept is similar to the Williams' algorithm presented in chapter two but the method to select the unwanted points is different. This algorithm handles the inflection point problem and is simple to implement. The algorithm is detailed in the following.

To eliminate excessive points along the contour, it is necessary for the user to provide a tolerance measure. This tolerance measure is used to define an "exclusion region" to determine superfluous points so that they can be eliminated later. The exclusion region refers to the area bounded by the two circles centered at both end points and the dotted lines tangent to the circles as shown in Figures 3.3 and 3.4. The process is started at one end of the contour and proceeds with the following steps:

1). Start at P_m,

see if P_{m+1} is close enough to line connecting P_m and P_{m+2} (denoted by P_mP_{m+2}).

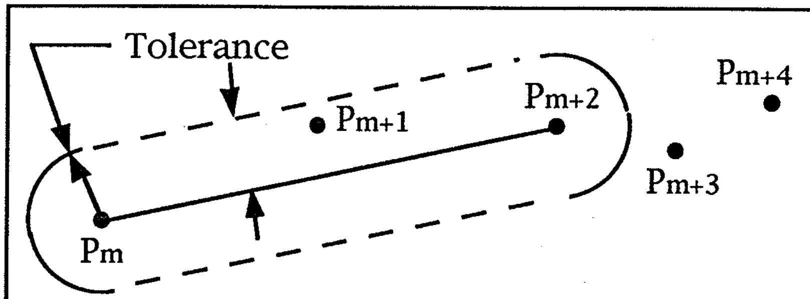


Figure 3.3

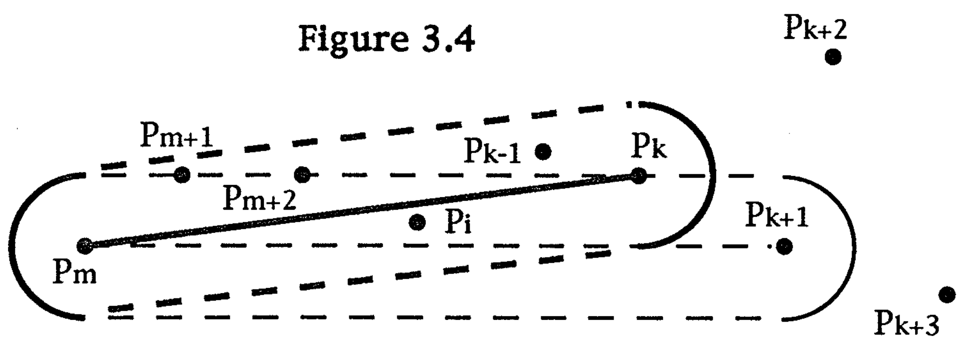
see if P_{m+1}, P_{m+2} are close enough to line P_mP_{m+3}.

$\vdots \quad \vdots \quad \vdots$

see if P_j, $j = m+1, m+2, \ldots, k-1$ is close enough to line P_mP_k.

2). Find the largest k in step 1). As shown in Figure 3.4, if the exclusion region is extended to include P_{k+1}, P_{k-1} would have to be thrown outside of the region and thus P_k is the largest possible k.

Figure 3.4



3). Eliminate P_j, j = m+1, m+2, ..., k-1 and retain P_m and P_k (or the line P_mP_k).

4). Start next line at P_k and the process repeats from step 1) until the input is exhausted.

When all these steps are completed, the points that are connected with line segments are the required points and they can be used to generate the tool path.

Note that a special case occurs when only two points remain in step one; in this case the process is completed by defining a line segment joining them and the subsequent steps will not be performed.

The entire data-reduction process is simple and only two sequential subprocesses are needed as shown in the Data Flow Diagram of Figure 3.5.

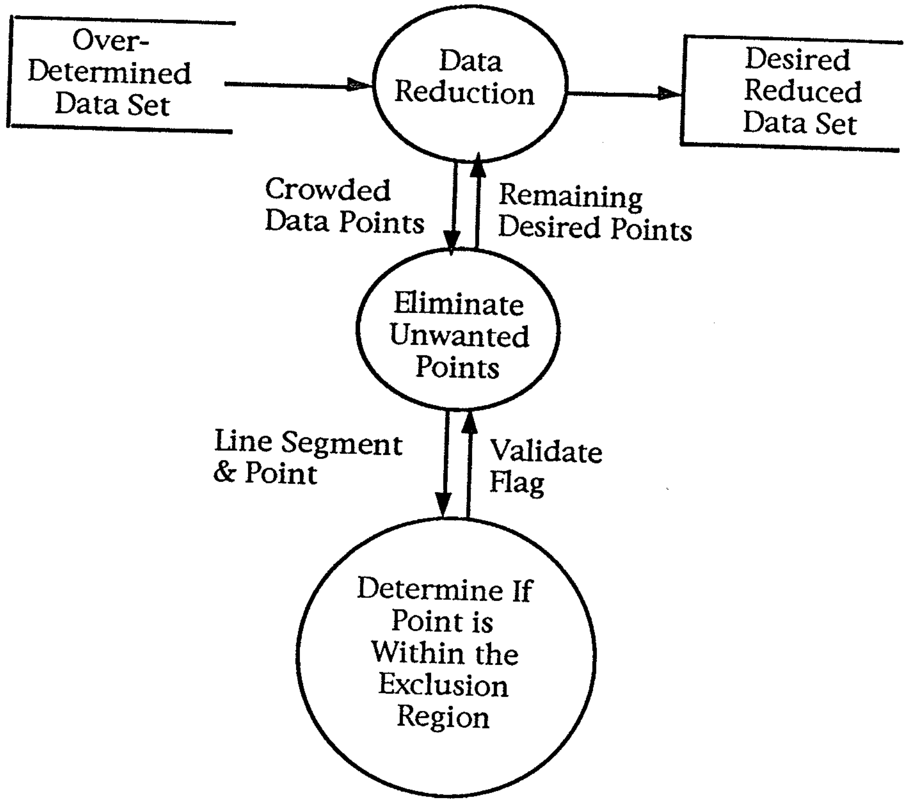


Figure 3.5 Data Reduction

3.3 Parametrization of contours and the interpolation of grid points for the construction of a B-spline surface.

When the input data is not aligned properly and/or the number of co-planar points are different between contours due to random data-acquisition or as a result of data-reduction, it can not be used for parametric surface construction. It is necessary to either relocate these existing points or redefine a new set of data points so that they are properly aligned and well ordered. Relocation of existing points is not suitable because it may cause loss of the only information from the original contour unless an appropriate mechanism is implemented to ensure that the movement of the points is along the contour curve. On the other hand, redefinition of a new set of well ordered and properly aligned data points can be done using spline interpolation. The unaligned and unmatched input data points are used to construct a parametric cubic spline which can be used to interpolate a new set of data points for each planar contour. This new set of data points is aligned with the same number of points for each planar contour.

In this project, a parametric periodic cubic spline is chosen for the interpolation of new data. An interpolating cubic spline has the same C^2 continuity properties as guided cubic B-splines but does not have the local control feature. The definition of a parametric cubic curve is :

$$\delta(t) = At^3 + Bt^2 + Ct + D;$$

A parametric cubic spline is defined by number of cubic segments connected together such that they are continuous at the end points. Hence an n point parametric cubic spline can be defined as :

$$\delta_i(t) = A_i t^3 + B_i t^2 + C_i t + D_i, \text{ where } 0 \le i \le n\text{-}1.$$

As the planar contour representing the cross section of the object being built is a closed curve, grid points interpolated from this contour must be distributed around the object. One way to do this is to parametrize the contour using angular measurement α as the parameter with the origin inside the contour. The cubic spline is then:

$$\delta_i(\alpha) = A_i\alpha^3 + B_i\alpha^2 + C_i\alpha + D_i;$$

with $0 \le \alpha \le 2\pi$ the spline forms a closed C^2 curve so $\delta_0(\alpha)$ and $\delta_{n-1}(\alpha)$ match in first and second derivatives at their joining point. A point Ω, interior to the contour lines, should first be established as the origin of the angular parameters. The line joining Ω and δ_0 defines the X-axis and thus the corresponding angle $\alpha_0 = 0$ is established. The interpolation of new grid points is then started using the above definition. In order to align the points between contours, a given set of angles, α_i, i = 0, ..., n-1, such that

$$\alpha_i = (2i\pi) / n, i = 0, 1, \ldots, n$$

is used to interpolate the same given number of points for all the contours. Hence the configuration of this new data set is aligned.

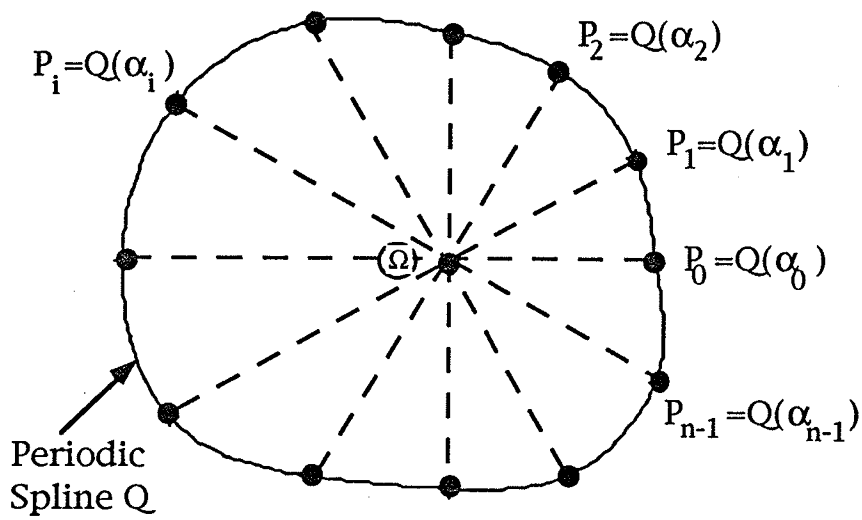


Figure 3.6 Use angular measurements as the parameters to interpolate points on a periodic spline

Figure 3.6 demonstrates the interpolation of new data points using angular measurements as the input parameters on a periodic cubic spline. It shows that the interpolated points are on the curve. With this method the corresponding polar angles of these interpolated points are not exactly equal to the input angles but they are very close. The following table shows the differences between the results of twelve interpolated points and their corresponding input parameters.

Point (X, Y)	Corresponding Polar Angle (radian)	Input Parameter (radian)
(1.8286 0.0000)	0.0000	0.0000
(1.6358 0.9474)	0.5249	0.5236
(0.9784 1.7104)	1.0512	1.0472
(0.0303 2.0393)	1.5559	1.5708
(-1.2387 2.1345)	2.0966	2.0944
(-2.4083 1.4237)	2.6077	2.6179
(-2.8880 0.0000)	3.1416	3.1416
(-2.4529 -1.4129)	3.6642	3.6652
(-1.2281 -1.9157)	4.1423	4.1888
(0.0303 -2.0268)	4.7273	4.7124
(1.0996 -1.8894)	5.2395	5.2359
(1.8462 -1.0661)	5.7595	5.7596

Data points given in polar coordinates are already in the desirable form of (α, r) where α is the angular parameter for the parametrization, provided that the origin is inside the contour line. The radial distance is given by r. To complete the parametrization, it is necessary to convert the points to the Cartesian form of (x, y) by:

$x = r * \cos(\alpha)$; and $y = r * \sin(\alpha)$;

The flow chart in Figure 3.7 illustrates this process such that the unaligned and unmatched data set is transformed to a new data set which is properly aligned and well ordered with the same number of points for each contour.

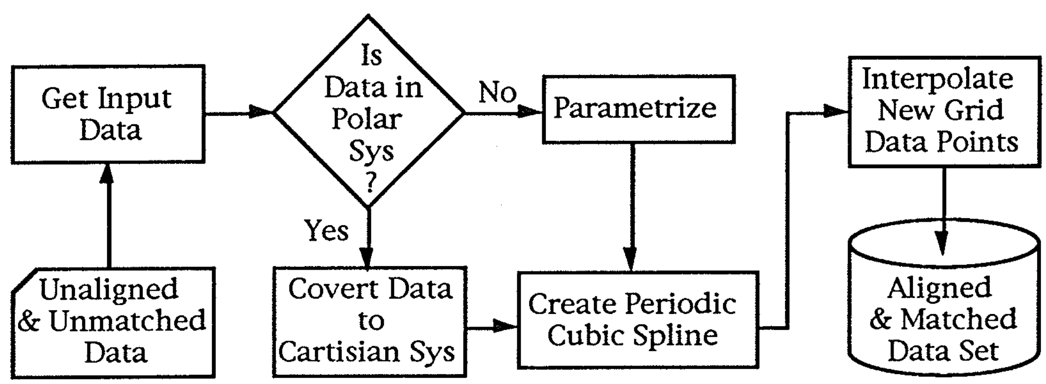


Figure 3.7 Interpolate New Points from Periodic Spline.

To summarize, if the given data of the contour is presented in polar form they should be converted to Cartesian form but are already parametrized in terms of angle. However, if the data is in Cartesian form they should be parametrized in terms of angle. After parametrization, cubic spline interpolation can be used to obtain properly aligned points on a well ordered grid in parametric space. These points can be used as the control vertices to construct a parametric surface in three dimensions.

3.4 Construction of bicubic B-spline surfaces to approximate the objects being built.

In the sequel, B-splines and B-spline surfaces will mean cubic B-splines and bicubic B-spline surfaces respectively. The final process of this project is to create the B-spline surface to approximate the shape of the object. In order to construct a well formed B-spline surface, the number of the control vertices in the rows and in the columns are critical. Each patch of the B-spline surface is constructed by 4 consecutive rows and 4 consecutive columns of points. That means a B-spline surface requires a minimum of 4 rows and 4 columns for a total of 16 points. The number of patches increase as the number of rows and/or columns increase, i.e.

$$P = (R - 3) \times (C - 3)$$

where P = number of patches, R = number of rows and C = number of columns.

As is shown in Figure 3.8, a B-spline surface composed of four contiguous patches is created by the 25 points of a square grid of 5 rows and 5 columns. The four patches are distinguished by different patterns. The corresponding corner points indicate the 4 by 4 grid that created the patch. For example, the corresponding corner points for the first patch are labelled 1, 4, 19 and 16. That means the first patch is defined by the points bounded by these four corners i.e. the points 1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19.

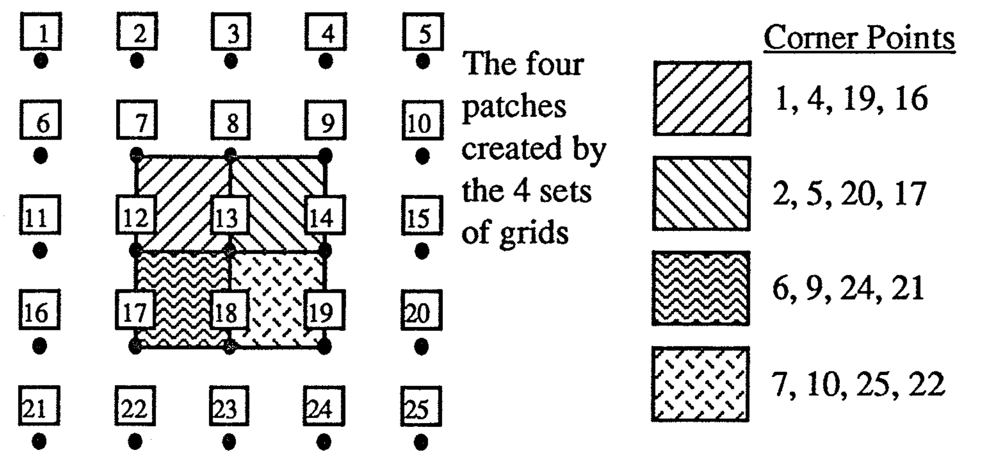


Figure 3.8 Four patches created by 25 points.

The process to create the B-spline surface is divided into several subprocesses from obtaining input data to output of the complete surface.

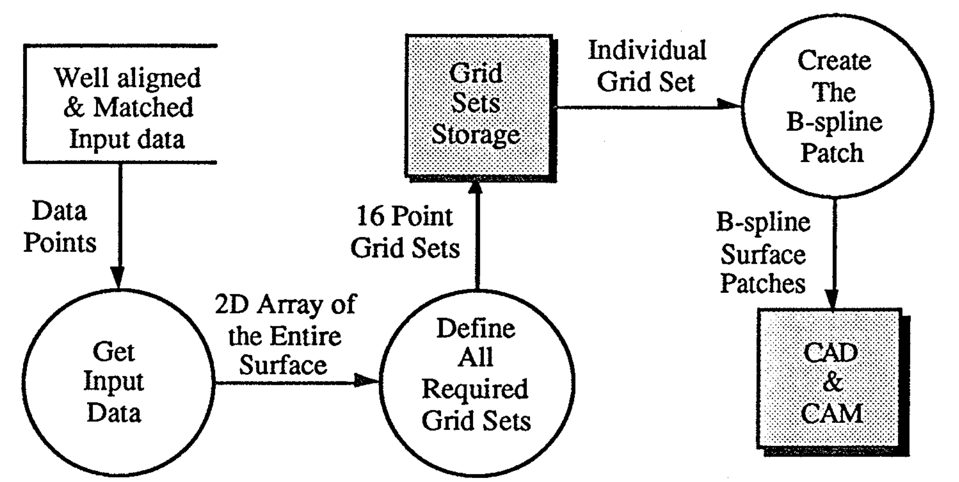


Figure 3.9 B-spline surface creation.

As shown on the data flow diagram in Figure 3.9 the process starts by entering in a set of well ordered and properly aligned grid of points from a data file. The data points are then arranged as a two dimensional array such that each row corresponds to a planar contour and each column corresponds to a set of aligned points, one from each contour. Since each contour is a closed curve and the surface representing the object should be closed, the first three consecutive points are repeated after the last point P_{n-1} for convenience in implementation, so there are n+3 points for each row (where n is the number of points of a contour). If a contour is represented by the points P_0, P_1, P_2, ..., P_{n-1} then the corresponding row will contain the points P_0, P_1, P_2, ..., P_{n-1}, P_0, P_1, P_2. The second step is to define all possible grid point sets such that each set contains all of the intersection of 4 consecutive rows and 4 consecutive columns. Hence the set will contain a total of 16 points with 4 points per row and 4 points per column. Each grid set is then used as the control vertices to create one B-spline surface patche. The definition of parametric bicubic surface,

$$Q(s, t) = S \cdot M \cdot G \cdot M^T \cdot T^T, \qquad \text{for } 0 \le s, t \le 1,$$

can be used to generate a B-spline patch by replacing M with the B-spline's basis matrix [11] and G with the corresponding grid set. With appropriate values for s and t, any point on the patch can be interpolated.

After all the patches are defined, the construction of the surface is completed. This surface features local control as well as arbitrary point interpolation. When a control vertex is modified a maximum of 16 patches i.e. 4 along the row's direction and 4 along the column's direction, are affected depending on the location of the vertex. Points on the surface can be interpolated by changing the values of the parameters s and t. Therefore this surface is suitable for design and modification of the object in the CAD application as well as for the interpolation of tool points to generate tool paths in the CAM process. With an appropriate number of points or splines defined along and across the surface, it can also display the represented object for visualization of the design. Indeed, the B-spline surface is very useful in CAD/CAM applications.

3.5 Summary.

In this chapter, a method to construct a three dimensional object is proposed. The suggested technique is detailed in three major phases. A data point reduction algorithm is presented to reduce excessive data points to minimize the load to the subsequent processes. The use of cubic splines to approximate planar contours allows the interpolation of a new set of well ordered and properly aligned grid of data points in case the input data points are not properly aligned and/or ordered. Finally, a B-spline surface is created using these well ordered grid points as the control vertices to approximate the shape of the object. This representation can then be used for various CAD/CAM applications.

The data reduction scheme used in this project was presented and implemented by M.Yeung and D.Walton[4]. It is based on the "longest line"[16] method. In this scheme, an exclusion region between two distance points is defined to test the intermediate points between them. The implementation of this algorithm is simple and fewer processes are required.

In order to align coplanar data points, a periodic cubic spline is used to form a planar contour for each set of coplanar points and these splines in turn are used to interpolate a new set of data points which will be aligned. The periodic cubic spline is chosen because of it gives a smooth approximating closed curve and the original data points are interpolated.

A bicubic B-spline surface is used to approximate the object in order to provide the local control capability as well as to maintain the C^2 continuity between patches for the CAD process. On the other hand, the parametric nature of this surface allows the easy interpolation of surface points for CAM processes. These combined features make the design of three dimensional object easier and more convenient.

This concludes the discussion of the method used for the project described in this thesis. The implementation of this method is discussed in the next chapter. A sample object will be constructed.

CHAPTER IV
IMPLEMENTATION OF THREE DIMENSIONAL OBJECT REPRESENTATION WITH PLANAR CONTOURS AND PARAMETRIC SURFACES

4.1 Introduction.

To apply the techniques described in chapter three to construct a three dimensional object from input data points, a special computer system is required. There are many commercially available hardware and software systems that can be used for the implementation but in order to maximize efficiency, several criteria should be considered in the selection of the system. For instance, in the process of constructing a three dimensional object using coplanar contours and parametric surfaces, intensive and complicated mathematical calculations are involved, therefore an efficient and comprehensive mathematical library and/or math-coprocessor would be required to speed up the processing time. Secondly, to display the surfaces that represent the object realistically, high resolution and extensive graphics capability are needed. The selected computer system should be equipped with an enriched graphics library and a set of effective and powerful graphics display hardware components such as a graphic processor, high resolution monitor, etc.

In the following sections, the hardware and software chosen for the implementation are described and the implementation of the functions and procedures such as data reduction, interpolation of new data points with parametrization, and the construction of a B-spline surface to represent the object are detailed.

4.2 Hardware and software for the implementation.

With the mathematical and graphical constraints, the Silicon Graphics Personal IRIS 4D/35 computer running IRIX 3.3.1 (Silicon Graphics' UNIX) is chosen for the implementation of this project. Configuration of this Silicon Graphics system includes :

- 33MHZ 32 bit RISC CPU.
- 33MHZ floating point co-processor.
- 27MIPS processor.
- 24 color bitplanes.
- 16MB RAM.
- 19" 1280x1024 High resolution color monitor.

Development software includes 4 Sight windowing system, and the MIPS-C compiler which includes a comprehensive math library as well as an extensive graphics library. Graphics display and manipulation such as rotation, clipping and scaling are performed in a combination of custom VLSI circuits, conventional hardware, firmware, and software[17]. The development environment also includes the 4 Sight windows as well as the window-based standard UNIX debugger "Edge" from Silicon Graphics.

4.3 Data point reduction procedures.

To implement the "Tolerance dependent data-point elimination" for a planar contour, several functions and procedures are defined to perform the operations. Functionality of these functions and procedures correspond to the processes defined in section 3.2 on the determination of unwanted points. The sequence of execution reflects the steps described in that section. Since the data points to be reduced are a set of coplanar points. Reduction of the data thus reduces to a problem in two dimensional geometry. For simplicity, in this illustration, assume the points are in the Z-plane. The following seudo code describes the implementation of this process for n coplanar points.

```
/* Start Process */
start_at = 0;
Insert P_start_at into the resulting array;
end_at = 3;
done = FALSE;
WHILE NOT done DO
        WHILE end_at ≤ n AND
                Within_Exclusion_Region ( start_at, end_at, tolerance ) DO
                end_at = end_at + 1;
        done = end_at > n;
        Insert P_end_at-1 into the resulting array;
        start_at = end_at  - 1;
        end_at = end_at +1;
END WHILE
/* End of Process */
```

Within_Exclusion_Region is a boolean function that returns TRUE if there isn't any points between P_{start_at} and P_{end_at} or all the points P_i, start_at < i < end_at are inside the exclusion region of the line segment joining P_{start_at} and P_{end_at}. It returns FALSE if any one or more of these points P_j, $j \in i$ is outside the exclusion region. In order to implement this function the location of P_i and its distance to the line joining P_{start_at} and P_{end_at} are needed to determine if it is within the exclusion region or not. To do this, the following steps can be taken[16].

1). Define the line segment L jointing P_{start_at} and P_{end_at} as

$$L = (1 - t)P_{start\_at} + t\,P_{end\_at}$$

2). Solve for t by

$$t = \frac{(P_i - P_{start\_at}) \bullet (P_{end\_at} - P_{start\_at})}{(P_{end\_at} - P_{start\_at}) \bullet (P_{end\_at} - P_{start\_at})}$$

3). Define the distance D from P_i to L

$$D = \begin{cases} \left\| (P_i - P_{start\_at}) \times \dfrac{(P_{end\_at} - P_{start\_at})}{\| P_{end\_at} - P_{start\_at} \|} \right\| & 0 \le t \le 1 \\ \| P_i - P_{start\_at} \| & t < 0 \\ \| P_i - P_{start\_at} \| & t > 1 \end{cases}$$

If D > Tolerance, P_i is outside the exclusion region of the segment L, otherwise it lies inside the exclusion region and can be eliminated.

The entire process structure is illustrated in the structure chart of Figure 4.0.

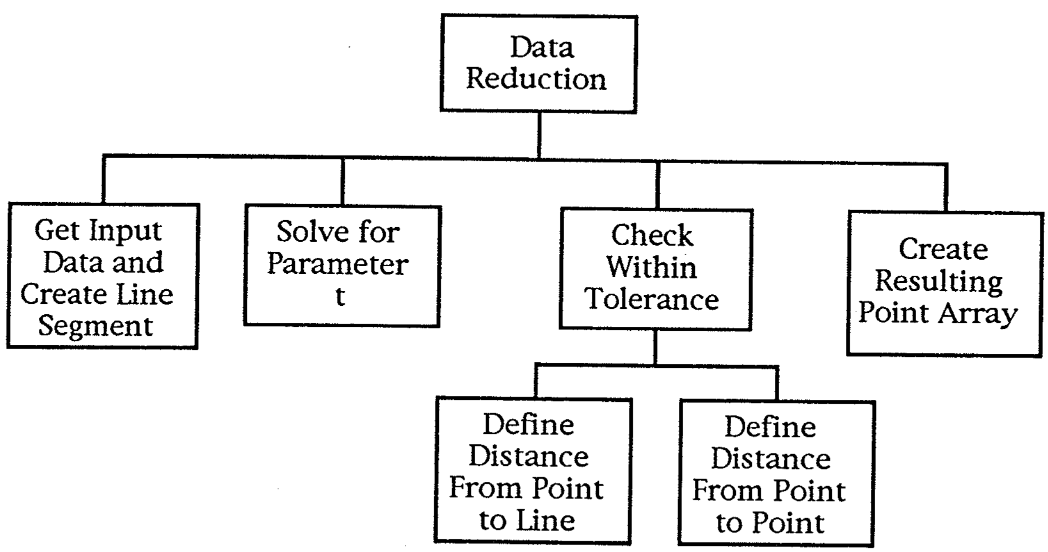


Figure 4.0 Struture chart for data-reduction process

4.4 Interpolation of data points for the construction of bicubic B-spline surfaces.

The next task after the process of data-reduction is to generate a set of ordered and aligned data points from the modified coplanar contours. For each contour, a constant number of new data points will be interpolated and aligned from contour to contour. The steps of the process are described below.

4.4.1 Construction of a parametric periodic cubic spline from existing contour points.

To define the parametric periodic cubic spline[18], m data points will generate m spline segments. For segment i, the cubic equations are

$$X_i(u) = e_i(u - u_i)^3 + f_i(u - u_i)^2 + g_i(u - u_i) + h_i$$

$$Y_i(u) = a_i(u - u_i)^3 + b_i(u - u_i)^2 + c_i(u - u_i) + d_i$$

where u is the parameter. Since both $X_i(u)$ and $Y_i(u)$ are treated in the same way, therefore only $Y_i(u)$ is used to illustrate the definition of the spline. With the above equation $Y_i(u)$, then

$$Y_i(u_i) = y_i \qquad = d_i$$

$$Y_i(u_{i+1}) = y_{i+1} \qquad = a_i\Delta u_i^3 + b_i\Delta u_i^2 + c_i\Delta u_i + d_i$$

because there are four coefficients to be determined, two other constraints are needed to completely determine a particular $Y_i(u)$. If the first derivative y' and second derivative D of Y(u) at each knot y_i are used as constraints, then

$$Y_i^{(1)}(u_i) = y'_i \qquad = c_i$$

$$Y_i^{(1)}(u_{i+1}) = y'_{i+1} \qquad = 3a_i\Delta u_i^2 + 2b_i\Delta u_i + c_i$$

$$Y_i^{(2)}(u_i) = D_i \qquad = 2b_i$$

$$Y_i^{(2)}(u_{i+1}) = D_{i+1} \qquad = 6a_i\Delta u_i + 2b_i$$

The four coefficients can be determined in terms of the function values and the second derivatives

$$a_i = (D_{i+1} - D_i) / 6(\Delta u_i),$$

$$b_i = D_i / 2,$$

$$c_i = \Delta y_i / \Delta u_i - \Delta u_i(D_{i+1} + 2D_i) / 6,$$

$$d_i = y_i.$$

To define all the D_i, conditions at each knots y_i are used, i.e.

$$Y_{i-1}(u_i) = y_i, \qquad Y_{i-1}^{(1)}(u_i) = Y_i^{(1)}(u_i),$$

$$Y_i(u_i) = y_i, \qquad Y_{i-1}^{(2)}(u_i) = Y_i^{(2)}(u_i).$$

Furthermore, because it is a closed spline, for m knots (where $u_m = u_0$ as the knot sequence is cyclic) the end conditions are

$$Y_0(u_0) = Y_{m-1}(u_m), \quad Y_0^{(1)}(u_0) = Y_{m-1}^{(1)}(u_m), \qquad Y_0^{(2)}(u_0) = Y_{m-1}^{(2)}(u_m)$$

and the index i of the above four conditions becomes modulo of (m + 1).

The system of equations that define the D_i's with non-uniform knot sequence is then[18]

$$M D = Y$$

where

$$M = \begin{bmatrix} 2(\Delta u_{m-1}+\Delta u_0) & \Delta u_0 & & \cdots & & \Delta u_{m-1} \\ \Delta u_0 & 2(\Delta u_0+\Delta u_1) & \Delta u_1 & & \cdots & \\ & \Delta u_0 & 2(\Delta u_1+\Delta u_2) & \Delta u_2 & \cdots & \\ & & \cdots & \cdots & \cdots & \\ \Delta u_{m-1} & & & \Delta u_{m-2} & 2(\Delta u_{m-2}+\Delta u_{m-1}) & \end{bmatrix}$$

$$D = \begin{bmatrix} D_0 \\ D_1 \\ \cdots \\ \cdots \\ D_{m-1} \end{bmatrix}, \qquad Y = \begin{bmatrix} 6(\Delta y_0/\Delta u_0 - \Delta y_{m-1}/\Delta u_{m-1}) \\ 6(\Delta y_1/\Delta u_1 - \Delta y_0/\Delta u_0) \\ \cdots \\ \cdots \\ 6(\Delta y_{m-1}/\Delta u_{m-1} - \Delta y_{m-2}/\Delta u_{m-2}) \end{bmatrix}$$

To solve this Tridiagonal-circulant system, seven vectors are used, they are :

q[m] = The main diagonal.

r[m-1] = The lower band.

s[m-1] = The upper band.

g[m] = The vector Y.

e[m-1] = The last column of M, note that e[m] = a[m].

f[m-1] = The last row of M, note that f[m] = a[m].

x[m] = The resulting vector D (the derivatives).

The algorithm[18] is illustrated by the following pseudo code.

```
/* Start Process */

Initialize q,r,s,e, f and g to their repective values.

/* Process the first m-1 rows */

FOR i = 1 to m-2 DO

        scalor1 = r_{i-1} / q_{i-1};

        q_i = q_i - s_{i-1} * scalor1;

        e_i = e_i - e_{i-1} * scalor1;          /* Update the last column */

        scalor2 = f_{i-1} / q_{i-1};

        f_i = f_i - s_{i-1} * scalor2;  /* Update the last row */

        q_{m-1} = q_{m-1} - e_{i-1} * scalor2;    /* Update last element of q */

        g_i = g_i - g_{i-1} * scalor1;

        g_{m-1} = g_{m-1} - g_{i-1} * scalor2;

END FOR

/* Complete the last row */

scalor1 = f_{m-2} / q_{m-2};

q_{m-1} = q_{m-1} - e_{m-2} * scalor1;
```

$g_{m-1} = g_{m-1} - g_{m-2} * scalor1;$

/* Solve for x */

$x_{m-1} = g_{m-1} / q_{m-1};$ /* Last element of x */

$x_{m-2} = (g_{m-2} - e_{m-2} * x_{m-1}) / q_{m-2};$

FOR i = m-3 DOWNTO 0 DO

$x_i = (g_i - s_i * x_{i+1} - e_i * x_{m-1}) / q_i;$

/* End of Process */

To complete the spline, the m sets of coefficients of the m segments can now be defined with the following

/* Evaluate the first m-1 sets of coefficients */

FOR i = 0 TO m-2 DO

$a_i = (D_{i+1} - D_i) / 6(\Delta u_i),$

$b_i = D_i / 2,$

$c_i = \Delta y_i / \Delta u_i - \Delta u_i(D_{i+1} + 2D_i) / 6,$

$d_i = y_i.$

END FOR

/* Complete the last set of coefficients */

$a_{m-1} = (D_0 - D_{m-1}) / 6(\Delta u_{m-1}),$

$b_{m-1} = D_{m-1} / 2,$

$c_{m-1} = \Delta y_{m-1} / \Delta u_{m-1} - \Delta u_{m-1}(D_0 + 2D_{m-1}) / 6,$

$d_{m-1} = y_{m-1}.$

The parametric periodic cubic spline is completely defined. Any point on the spline can then be interpolated with an appropriate parameter u.

4.4.2 Interpolation of new data points from the periodic cubic spline.

In order to interpolate a new well ordered set of data points, aligned from contour to contour, from the periodic cubic spline, parametrization with angular measurement as a parametric value is used. The user provides the number of new points desired per contour to be interpolated. To start the process, the point of rotation or origin is established inside the closed spline. The values of the angular parameters for point interpolation will be referenced about this point during the interpolating process. Since the sample data is obtained via a laser scanner in a circular orbit, the point of rotation is already inside every contour; it is the XY-origin on the Z-plane i.e. (0, 0, z). The next step is to define the corresponding angular orientations for the existing knots. To do this, a function Find_angle(P) based on the intrinsic arctan function which returns the corresponding polar angle δ (where $0 \leq \delta \leq 2\pi$) of the point P is used. For every knot P_i the angular orientation δ_i=Find_angle(P_i) is defined.

In order to interpolate the new point q_i at α_i which will be aligned with the corresponding point g_i at β_i (i.e. $\alpha_i = \beta_i$) of the neighboring contour line, the x-axis is used as the basis of α's; i.e. $\alpha_0 = \beta_0 = \ldots = 0$ and thus all subsequent points interpolated will be aligned from contour to contour; i.e. $\alpha_i = \beta_i = \ldots$ and so on. Since the starting point of every spline is already aligned to each other as the result from data acquisition, it is convenient to set the starting point to α_0. In the event the starting point for each spline is not aligned to each other, a common basis of α's for all the splines must be defined. To do that, the spline segment which is intersected by the ray emanating from the center of rotation with direction α_i should be located. To find the correct spline segment the corresponding orientations $\delta_0, \delta_1, \ldots, \delta_m, \delta_0$ of the existing points are searched until α_i lies in the interval bounded by δ_j and δ_{j+m1}, where j = j mod (m+1). When the segment is found, two cases can occur.

(i). $\delta_j < \delta_{j+m1}$ and $\delta_j \le \alpha_i < \delta_{j+m1}$;

e.g. $\delta_j = 0$, $\delta_{j+m1} = \pi / 3$ and $\alpha_i = \pi / 4$.

(ii). $\delta_j > \delta_{j+m1}$ and ($\delta_j \le \alpha_i < 2\pi$ or $0 \le \alpha_i < \delta_{j+m1}$);

e.g. $\delta_j = 11\pi / 6$, $\delta_{j+m1} = \pi / 4$ and $\alpha_i = \pi / 5$.

For case (i), the parameter α_i is not changed, but for case (ii), there are two possible values for α_i. In case (ii), α_i is not changed if $\delta_j \le \alpha_i < 2\pi$, and $\alpha_i = \alpha_i + 2\pi$ if $0 \le \alpha_i < \delta_{j+m1}$. The parameter α_i can then be substituted into the cubic to interpolate the new point such that

$$q_i(\alpha_i) = a_j + b_j\alpha_i + c_j\alpha_i^2 + d_j\alpha_i^3$$

The following pseudo code shows how this is done.

```
/* Start Process */
case = 0;        /* Needed only when the spline points are not aligned */
j = 0;
WHILE the interval ( δj , δj+m1 ) that contain αi is not found DO
        j = ( j + 1 ) mod m;
Determine the case as above and update αi;   /* Needed only when the spline
                                                 points are not aligned */
Interpolate the point qi.
/* End of Process */.
```

Finally to interpolate the number of new data points that the user specified for the contour, a simple FOR loop is used.

```
α = 0;
increment = 2π / n;    /* Where n is the number of new points */
FOR i = 0 TO n - 1
```

Interpolate q_i with α;

$\alpha = \alpha$ + increment;

END FOR

With this parametrization, the coordinates of a point are functions of α as $X_i(\alpha)$ and $Y_i(\alpha)$ and the corresponding polar angle of the point interpolated is very close to the input angular parameter as stated in section 3.3. Another possible way to set up the parametrization is let the radius r be the function of α, such that $r_i = R_i(\alpha)$. With this method, the coordinates of the point are obtained by

$x_i = r_i * \sin \alpha,$

$y_i = r_i * \cos \alpha.$

The point interpolation is the same as the process described above, but only one set of coefficients is required for r_i instead of two sets for x_i and y_i and the angle interpolated will be exactly equal to the input angular parameter. This method was implemented and tested by a colleague but according to the report, the result was somehow unsatisfactory under certain conditions. Further study on this method may be required.

4.4.3 Generation of grids for the construction of bicubic B-spline surfaces.

The last task for this section is to complete the interpolation of new data points for the final phase of this project in which the bicubic B-spline surfaces to represent the object being built, are constructed. To interpolate these points from m coplanar curves Ci, i = 0, ..., m-1, a simple FOR loop is applied.

FOR i = 0 TO m-1 DO

Interpolate n points from C_i using the technique of parametrization with angular measurement as parameter;
/* where n is number of points per curve specified by the user */

END FOR.

To sum up the implementation of interpolation of new data points for surface construction, the following structure chart illustrates the process models and their hierarchical relations.

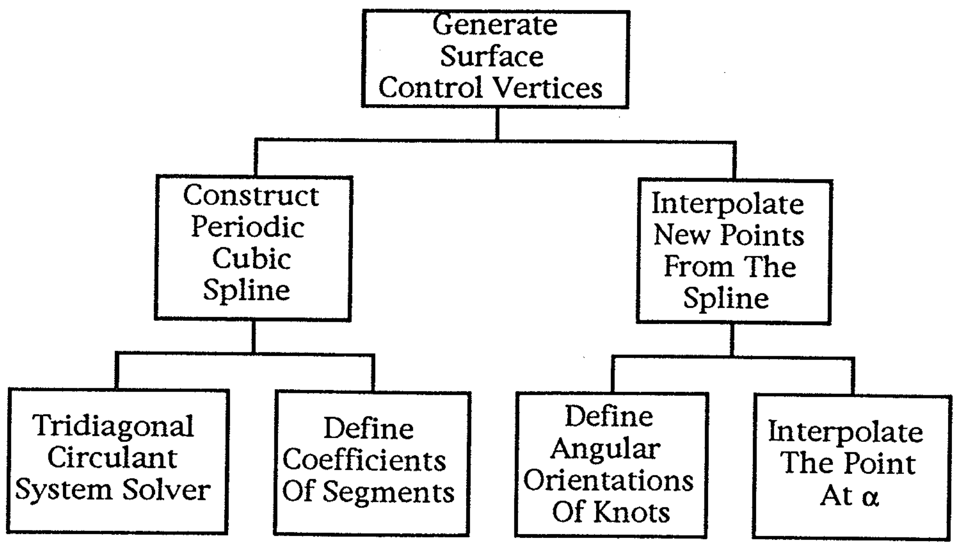


Figure 4.1
Generation of Control Vertices for Surface Construction

and the pseudo codes for the process of surface grid interpolation for the m coplanar curves is now listed.

/* Start Process */

Obtain the number, n, of new points per contour required from the user;

Let P[m,n] be the surface grid array.

FOR i = 0 TO m - 1 DO

Fit a periodic cubic spline Q to contour C_i;

```
        increment = 2π / n;
        α = 0;
        FOR j = 0 TO n - 1 DO
                Interpolate point P_i,j from Q with angular parameter α;
                α = α + increment;
        END FOR
END FOR
/* End of Process */
```

4.5 Construction of bicubic B-spline surfaces to represent the object.

The final process of this project is to construct a bicubic B-spline surface with the newly generated surface grids or control points. To construct a nice and smooth bicubic B-spline surface patch, a minimum of 16 points, arranged in an ordered configuration, are required. The proper configuration of the 16 points for the patch is a 4 x 4 matrix. The equation of the patch is

$$Q(s,t) = S \cdot M \cdot G \cdot M^T \cdot T^T \qquad (4.1)$$

where

$S = [\ s^3\ s^2\ s\ 1\ ]$; s is the parameter for the u direction of the patch.

$T = [\ t^3\ t^2\ t\ 1\ ]$; t is the parameter for the v direction of the patch.

G is the geometric constraint with 16 elements in a 4 x 4 matrix.

$$G = \begin{bmatrix} P_{i,j} & P_{i,j+1} & P_{i,j+2} & P_{i,j+3} \\ P_{i+1,j} & P_{i+1,j+1} & P_{i+1,j+2} & P_{i+1,j+3} \\ P_{i+2,j} & P_{i+2,j+1} & P_{i+2,j+2} & P_{i+2,j+3} \\ P_{i+3,j} & P_{i+3,j+1} & P_{i+3,j+2} & P_{i+3,j+3} \end{bmatrix}$$

In this case the constraint elements $P_{k,q}$; $k = 1, \ldots, 4$; $q = 1, \ldots, 4$, are the control points which the patch is based on.

M is the basis matrix. For a bicubic B-spline surface it is defined statically as[11]

$$M = \begin{bmatrix} -1/6 & 3/6 & -3/6 & 1/6 \\ 3/6 & -6/6 & 3/6 & 0 \\ -3/6 & 0 & 3/6 & 0 \\ 1/6 & 4/6 & 1/6 & 0 \end{bmatrix}$$

For the construction of a surface patch or the interpolation of surface point on the patch, the following steps are implemented.

- Set up the geometric constraint matrix G with the input control points.
- Multiply T on the right by M, take the transpose of the result, multiply it on the left by G and call this result A.
- Multiply A on the left by M, and finally multiply this result on the left by S.

To complete the construction of the object, many patches are generated and connected together to form the entire surface of the object. In order to generate these patches, the well ordered data set is subdivided into many subsets. Each subset consists of 16 control points configured as an array of 4 rows and 4 columns. The criteria to select points for the patch was described in Section 3.4 of Chapter three. To implement the selection of points and the construction of these patches, several FOR loops are used.

To prepare the process, several variables are defined.

- Let P[r,c] be the input control points consisting of r contours or rows and c points per contour or column. Note that the first three points of the contour is repeated at the end of the contour. See section 3.4 of chapter three for details.
- Let G be the 4 x 4 geometric constraint matrix.

```
/* Start Process */
FOR i = 0 TO r - 4 DO
        FOR j = 0 TO c - 4 DO
                FOR k = i TO i + 3 DO
                        FOR m = j TO j + 3 DO
                                G[k-i,m-j] = P[k,m];
                Define a surface patch with G and the predefined M, S and T as
                described.
        END FOR.
/* End of Process */
```

The following structure chart sums up the surface creation process and thus concludes this section.

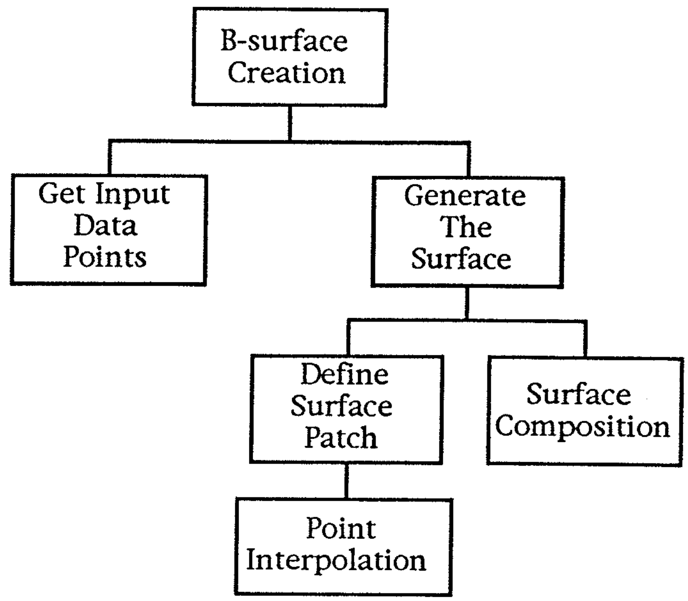


Figure 4.2 B-surface creation

4.6 Testing with a sample object.

To test this system for the modelling of a three dimensional object from closed planar contours and parametric surfaces, a sample part is produced. The sample part is a component of a prosthetic product produced by a prosthetic and orthotic product manufacturing company.

The sample data of this part is captured by a rotational laser scanner from the actual part as described in Section 2.1 of Chapter two. The format of the data is a set of closed planar contours that are 9.378 mm apart. There are a total of 35 contours to cover the portion of the part that is to be built. Each contour is defined by 256 points. Consecutive points along each contour line are separated from each other by $\pi/128$ radians. In this example, the center of rotation is also the origin of the coordinate system for the data points. Four values were provided for each point, namely the x, y and z coordinates in millimeters and the oriented angle relative to the x-axis in radians.

x	y	z	α

Representation of a point

The data is read into the system one contour at a time. Data reduction is performed with a specified tolerance to reduce the number of points of the contour. A periodic cubic spline is fitted onto the remaining points and then a fixed number of new points are interpolated using parametrization. These new data points are then written to a new data file. This process is repeated until all the contour lines are processed.

Finally, a B-spline surface is constructed based on this new set of data to represent the component. The resulting component appears to be similar to the actual component but with a much smoother surface. The image as displayed on a computer screen is shown in Figure 4.3.

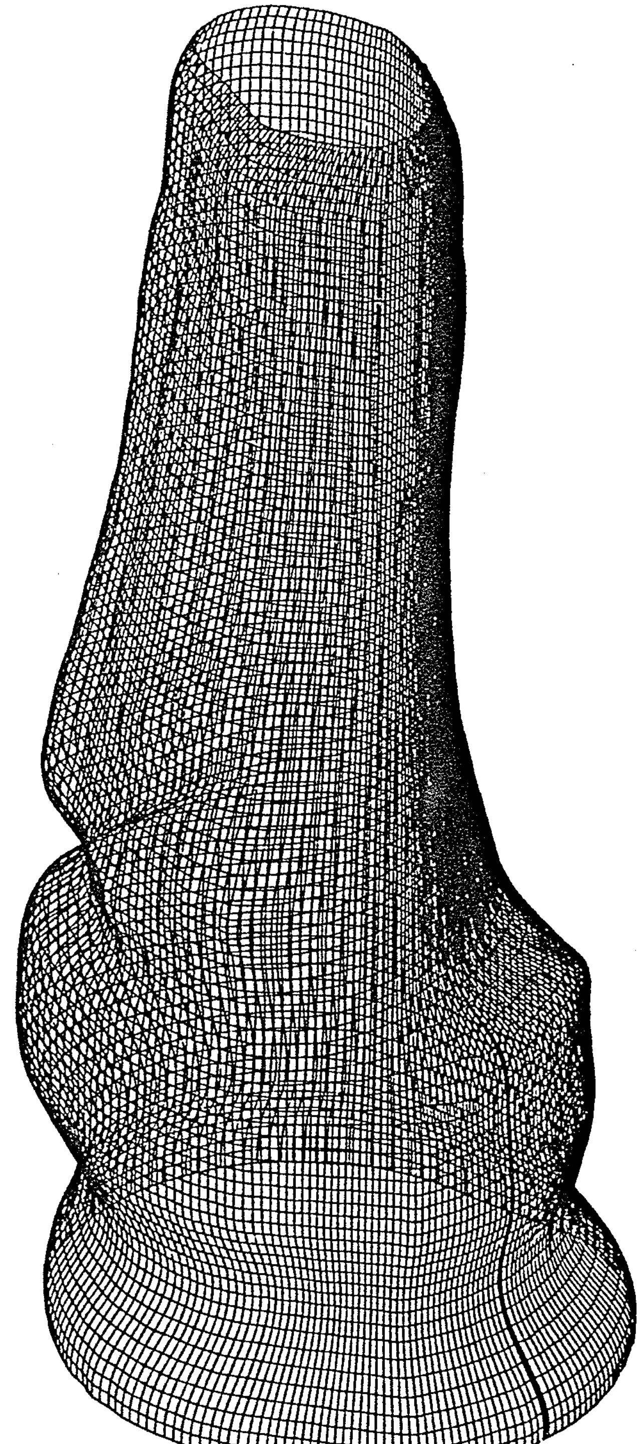


Figure 4.3 The sample object for testing.

4.7 Summary.

The construction of a three dimensional model of an object, using planar contours and parametric surfaces, is described in this chapter. The first five sections of the chapter are organized to reflect the major functions of the system, hardware and software requirements, data reduction, interpolation of surface grids, construction of a B-spline surface to represent the object, and testing with a sample object.

In order to deal with the extensive mathematical computation and the graphics requirement, a UNIX based Personal IRIS 4D/35 computer from Silicon Graphics was chosen for the implementation. It includes a high speed floating point co-processor, a powerful color graphics processor and a high resolution monitor. The MIPS-C compiler, with its mathematics and graphics libraries, was selected as the software development platform for the system.

The scheme of "Tolerance dependent data-point elimination" is used for the implementation of data reduction to reduce the excessive data points for each contour. The result of this process is a set of data points which is reduced and is within the specified tolerance from the original contour.

Three subtasks are performed sequentially to interpolate a new set of data points for the construction of bicubic B-spline surfaces. For each reduced contour, a periodic cubic spline parametrized with angular measurement is fitted onto the remaining points. That spline is used to interpolate a new set of contour points which are equally spaced by a fixed angular distance; the center of rotation for this parametrization is located inside the contour. This process is repeated until all the contours are processed and then the interpolation of control points for the construction of the bicubic B-spline surface is completed.

The final process of the system is to fit a bicubic B-spline surface based on the new data set to represent the object. The B-spline surface is composed of a number of surface patches. Each patch is constructed by 16 points that is configurated as an array of 4 rows and 4 columns. Selection of points is based on the criteria described in section 3.4 of Chapter three. When all the patches are built, the three dimensional object is fully represented.

A sample object was constructed to test the process scheme and the implementation. Data of the sample were obtained from a laser scanner. These data are reduced by the data reduction process, and the remaining points are used to interpolate a new set of data. A B-spline surface is then defined based on this new set of data to recreate the object. The resulting object appears similar to the original object but with a smoother surface.

This summarizes the implementation of three dimensional object representation with planar contours and parametric surfaces.

CHAPTER V
CONCLUSION

5.1 A brief summary.

In this thesis, several processes of the object production and reproduction in manufacturing industries and biomedical applications such as object design, data acquisition, data representation and object representation are presented and their related problems are also addressed. A number of solutions to these problems are discussed and evaluated against the requirements of the corresponding processes. The advantages and disadvantages of these solutions are analyzed and are later referenced in the project development of this thesis. In the second half of this thesis, a solution to reduce the size of the input data set and to construct the three dimensional object with closed planar curves and parametric surfaces is proposed and it's implementation is presented.

Two data acquisition systems that use laser scanners are described to emphasize the problems such as the over-determination that may arise during the data acquisition process. To reduce the problems caused by over-determination, several data-reduction schemes are examined and their inadequacies are emphasized. Other problems related to object design are the shape modification and surface representation. When a portion of the shape of the object is modified, the unaffected portions of the object should not be changed and the connection between the modified portion and the unaffected portions should maintain continuous with a smooth blend. Surfaces are commonly used for the representation of a three dimensional objects. Solutions to create surfaces with triangular polygons are presented but they do not deal with the shape modification problems.

To cope with the problem of over-determination, a specific data reduction scheme is suggested. This scheme overcomes the inflection point problem which other schemes, previously presented, suffer from. In dealing with unaligned data, periodic cubic splines based on the original data and the technique of parametrization are used to interpolate a new set of data which would be well ordered and aligned for subsequent applications. With a set of well ordered and aligned data points, a bicubic B-spline surface can be constructed to represent the object. The use of a parametric bicubic B-spline surface provides the local control feature while maintaining the C^2 continuity between patches.

The proposed solution was implemented on a Silicon Graphics computer system with a MIPS-C compiler as the development platform. The implementation of the suggested data reduction scheme is detailed step by step. Three subsections are dedicated to the implementation of the interpolation of surface grids which include construction of a periodic cubic spline, interpolation of points and the surface grid generation. The fitting of a bicubic B-spline surface is also implemented to represent the object being constructed. Finally, a sample object is constructed to test the proposed solution and the implementation of the system. The resulting model closely resembles the actual object but with a much smoother blend on the surface profile. The result of the testing is very satisfactory and concludes the project for this thesis.

5.2 Future prospects of planar contours and parametric surfaces.

In the proposed method, the powerful features of parametric surfaces and the convenience of using planar contours in the design of three dimensional objects were demonstrated. When integrated with other features of a CAD/CAM system, they can be utilized for other potential applications such as shape inspections, logical analysis and shape distortion compensation for the manufacturing industries. Some of these applications are already implemented with planar contours and parametric surfaces and are mentioned in the appendix, but most of them are still under research and development. These two geometric representations in CAD/CAM development and manufacturing industries have many important contributions and further research to improve their structures will definitely increase their acceptability and expand their applicability.

5.3 Acknowledgements.

I would like to thank Dr. Moe Barakat, Head of Robotics and Automation Section of National Research Council of Canada, Adjunct Professor of Department of Electrical Engineering, University of Manitoba who has co-supervised my work during my study for the degree of Master of Science. Also, a special thanks to Professor Dereck Meek of The Department of Computer Science, University of Manitoba who has contributed many constructive ideas for this thesis project.

Finally I would like to extend my very sincere appreciation to my academic supervisor Professor Desmond Walton of The Department of Computer Science, University of Manitoba who has guided me throughout the production of this thesis. His contribution to my work has provided the necessary stimulation for the success of this thesis and my graduate study at the University of Manitoba.

APPENDIX

1). VERIFOIL.

- a software system developed at National Research Council of Canada to analyze measured sections (blades, vanes) according to industrially specified airfoil features. It uses parametric periodic splines to represent various cross sections of the object in order to analyze the measured data.
- a portable program written in VAX FORTRAN 77, comprising approximately 10,000 lines of code and programming documentation. Verifoil was written in 1987-88 and has been used and improved since then.

Verifoil extracts and analyze foil characteristics such as:

- chord length
- chord thickness
- chord angle
- airfoil edge thickness (a means of indicating edge radii)
- form error
- station datum error (best-fit using least squares to MMC fits)
- station profile bisector
- edge radius quality status

for each section specified. The system interpolates a cubic spline through points measured at each section to define the measured shape.

Verifoil System Inputs can either be:

a) a coordinate-point file generated by a CMM, laser or other inspection device, and,

b) a file of parameters, including design nominal and tolerances, which control Verifoil's analysis,

c) a CAD file of the design splines and station planes. Verifoil currently is interfaced to a Unigraphics II CAD/CAM system, which is used to store the airfoil shape definition.

The system currently supports point input from:

- Brown & Sharpe VALMES11 format,
- DEA spectrum format,
- Diffracto Laser Gauge point format,
- MTC's Wahli51 probe,
- and a free format X,Y,Z point file.

Verifoil System Outputs:

Verifoil produces an analysis listing file, as well as graphics in the Unigraphics II CAD geometry file. The listing report includes a single page summary for each section analyzed, and a point-by-point report of each section's form errors. The graphical output created in the CAD geometry file includes a spline defining the measured shape, fit and form error vectors, and dimensions of each of the features measured on the blade section. Color is used to indicate quality status.

2). CLAMS.

- a software system developed at National Research Council of Canada to compare measured objects to their CAD models. Parametric bicubic surfaces are used to represent the shape and configuration of the object for the input of the CAD systems.
- a portable program written in VAX FORTRAN 77, comprising approximately 10,000 lines of code and programming documentation. It was developed in 1986-87, and is currently used at NRC .

Clams extracts and analyze two primary features:

- quality status of each point
- datum error (best-fit iteratively using least squares to MMC fits)

Clams System Inputs can either be:

a) a coordinate-point file generated by a CMM, laser or other inspection device. The system currently supports point input from:
   - Brown & Sharpe VALMES11 format,
   - DEA Spectrum format,
   - Diffracto Laser Gauge point format,
   - MTC's Wahli51 probe,
   - and a free format X,Y,Z point file.

b) a file of parameters, including design nominal and tolerances, which control Clams's analysis.

c) The CAD part file with surfaces defining the nominal shape. The program is interfaced to a Unigraphics II CAD/CAM system, which is used to store the design surfaces, and the graphics output.

In addition to parameters for analysis and measured points, Clams requires a definition of the design geometry. This can be given as a string of points defining the airfoil shape, or a spline definition in a CAD database. Clams currently is interfaced to a Unigraphics II CAD/CAM system, which is used to store the airfoil shape definition.

Clams System Outputs

Clams produces an analysis listing file, as well as graphics in the Unigraphics II CAD geometry file. The listing report includes a listing of the points measured, the individual errors, and a summary of the form errors. The graphical output created in the CAD geometry file includes the measured points, the error vectors (scaled) and the fit vectors (scaled). Color is used to indicate the quality status.

REFERENCES

1). "Measuring Machine System Overview", Spectrum Inspection System Operators Manual, Digital Electronic Automation, Inc. Livonia, MI. U.S.A., December 1988.

2). "Cyberware Technology Update", Cyberware Laboratory Inc. Monterey, CA. U.S.A., Application report. April, 1989.

3). El-Hakim S.F., "A Hierarchical Approach To Stereo Vision", Photogrammetric Engineering And Remote Sensing, Vol.55, No. 4, April 1989, pp.443-448.

4). Yeung M. and Walton D., "Data Point Reduction For NC Tool Path Generation On Over-determined Data Set", to be Presented at INCOM'92, 7th IFAC/IFIP/IFORS/IMACS/ISPE Symposium on Information Control Problems in Manufacturing Technology. May, 1992.

5). Rioux M., Blais F., Beraldin J.A. and Boulanger P., "Range Imaging Sensors Development At NRC Laboratories", from the Proceedings Of The Workshop On Interpretation Of 3D Scenes, TX, U.S.A., November, 1989.

6). Palser R., Jamieson R.,Sutherland J. and Skibo L., "Three-dimensional Lithographic Building From Volume Data Sets", from the Proceedings of The Canadian Medical and Biological Engineering Society Conference, June 1990.

7). McMaster R.B., "Automated Line Generalization". Cartographica Vol. 24, No. 2, 1987, pp. 74-111.

8). Reumann K. and Witkam A.P.M., "Optimizing Curve Segmentation In Computer Graphics", in the International Computer Symposium (A. Gunther, B.Levrat, and H.Lipps, Eds.), pp. 467-472, Amer. Elsevier, New York, 1974.

9). Williams C., "An Efficient Algorithm For The Piecewise Linear Approximation Of Planar Curves", Computer Graphics Image Process, Vol. 8, 1978, pp. 286-293.

10). Fowell R.A. and McNeil D.D., "Faster Plots By Fan Data-Compression", IEEE Computer Graphics & Applications, March 1989, pp. 58-66.

11). Foley J.D., van Dam A., Feiner S.K. and Hughes J.F., "Computer Graphics: Principles And Practice 2nd Edition", text book, published by Addison-Wesley Publishing Company, Inc. 1990, pp. 517-528.

12). Fuchs H., Kedem Z.M., and Uselton S.P., "Optimal Surface Reconstruction From Planar Contours", Communication of the ACM, Vol. 20, No.10, October 1977, pp.693-702.

13). Keppel E., “Approximating Complex Surfaces By Triangulation Of Contour Lines”, IBM Journal of Research and Development, Vol.19, January 1975, pp. 2-11.

14). Christianson H.and Sederberg T.W., “Conversion Of Complex Contour Line Definitions Into Polygonal Element Mosaics”, Computer Graphics, Vol.13, No.2, August 1978, pp. 187-192.

15) Barsky B.A. and Greenberg D.P., “Determining a Set of B-spline Control Vertices to Generate an Interpolating Surface”, Computer Graphics and Image Processing, Vol. 14, October 1980, pp. 203-226.

16). Meek D.S. and Walton D.J., “Several Methods For Representing Discrete Data By Line Segments”, Cartographica Vol. 28, No. 2, 1991, pp. 13-20.

17). “Introduction And System Overview Of IRIS-4D Series Workstation”, Graphics Library Programming Guide, Version 1.0, Document No. 007-1210-010, Silicon Graphics, Inc., Mountain View, CA. U.S.A. 1989.

18). Spath H., "Spline Algorithms For Curves And Surfaces", translation by W.D.Hoskins and H.W.Sager, Utilitas Mathematica Publishing Inc., Winnipeg, Canada.