2003

Multiresolution analysis as an approach for tool path planning in NC machining

Junhua Pang
Iowa State University

Recommended Citation

Pang, Junhua, "Multiresolution analysis as an approach for tool path planning in NC machining " (2003). *Retrospective Theses and Dissertations*. 736.
https://lib.dr.iastate.edu/rtd/736

Multiresolution analysis as an approach for tool path planning in NC machining

by

Junhua Pang

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:
Ranga Narayanaswami, Major Professor
Patrick Patterson
John Jackman
Abhijit Chandra
Thomas Barta

Iowa State University

Ames, Iowa

2003

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Graduate College
Iowa State University

This is to certify that the doctoral dissertation of

Junhua Pang

has met the dissertation requirements of Iowa State University

Signature was redacted for privacy.

Major Professor

Signature was redacted for privacy.

For the Major Program

TABLE OF CONTENTS

ABSTRACT

Wavelets permit multiresolution analysis of curves and surfaces. A complex curve can be decomposed using wavelet theory into lower resolution curves. The low-resolution (coarse) curves are similar to rough-cuts and high-resolution (fine) curves to finish-cuts in numerical controlled (NC) machining.

In this project, we investigate the applicability of multiresolution analysis using B-spline wavelets to NC machining of contoured 2D objects. High-resolution curves are used close to the object boundary similar to conventional offsetting, while lower resolution curves, straight lines and circular arcs are used farther away from the object boundary.

Experimental results indicate that wavelet-based multiresolution tool path planning improves machining efficiency. Tool path length is reduced, sharp corners are smoothed out thereby reducing uncut areas and larger tools can be selected for rough-cuts.

CHAPTER 1 INTRODUCTION

Tool path planning is regarded as one of the keys to the integration of CAD and CAM, and vital for the survival of manufacturing industry (Greenwood, 1989; Bedworth *et al.*, 1991). It is a fundamental task in NC machining. Planning is needed to execute both rough-cuts and finish cuts. In the rough-cut stage, the main goal is to remove material in the most efficient manner. In finish cutting, producing the desired surface finish and accuracy is the primary driving factor. Generation of the geometric profile of the tool path is the first stage. Interpolation methods are then applied to generate the NC code for machining.

1.1 Tool Path Generation Methods

Tool paths are the locus of cutter contact (CC) points, where the cutting tool makes contact with the design surface. The generation of tool paths has been a subject of a great deal of research in recent years.

Approaches to tool path generation can be broadly grouped into the following two categories.

- 2.5-dimensional (2.5D) machining. 2.5D machining is very important in the manufacturing of mechanical parts, moulds, and dies. 2.5D machining includes point-to-point, pocketing and contouring operations. Point-to-point operations like drilling and tapping do not give rise to geometrical problem, and are easy to be implemented in recent NC machines. Pocketing (a pocket is a steep walled recess) aims to form a depression in a workpiece by the removal of material within given boundaries. Contouring is planar cutting motion around a pre-defined part, which is similar to pocket machining an island without outer boundary walls. Tool path generation methods for pocketing and contouring include *zigzag method* and *contour-parallel method*, etc.. More detailed description about these methods will be presented later.

- 3-dimensional (3D) machining. Complex 3D machined surfaces can be produced because of the accurate tooling provided by the CNC machine. The tool path generation methods found in the literature for various surface representations are summarized as below.

Isoparametric curve method: This method is applicable only when the part surface is represented in the parametric form $\boldsymbol{S}$: $r(u,v)$ where u and v are the parameters of the surface $\boldsymbol{S}$. When using a CAD/CAM system to develop isoparametric tool paths, the user is prompted for the direction of tool motion, tolerances for curve approximations, and the tool step-over that specifies the distance between subsequent machining curves. The CAD/CAM system then proceeds to compute the required points that are either offset directly or projected onto the offset surface. The advantage of this method is the ease of calculating isoparametric curves on $\boldsymbol{S}$. However, The spacing between isoparametric lines, in general, will be non-uniform causing over machining or under machining. Bobrow (1985) proposes a method for computing the CL (cutter location) file directly from the isoparametric curves. Elber and Cohen (1994) have provided a three-axis adaptive isocurve ball-end mill machining algorithm that reportedly is gouge-free and eliminates the redundancy of the adjacent curves that are too close to one another.

Plane/Surface section curve method: This method is applicable for parametric, algebraic, or point-cloud representations of the surface. A set of parallel planes/surfaces (drive surfaces) is intersected with the design surface (Bobrow, 1985; Oliver and Huang, 1994). While any general surface may be used for intersections, planes and cylinders are most commonly used. The key advantage of this method is the machining of multiple adjoining surfaces in which curves from different surfaces are joined together into a single tool path. Disadvantages of this method are the computational burden involved and the numerical stability of surface intersection algorithms as intersecting surfaces approach tangency.

Projection curve method: This method is applicable for parametric, algebraic, or point-cloud representations of the design surface. In this method, the tool paths are created on a plane. The tool paths on the plane are then projected, in a given direction, onto the design surface (Choi *et al*, 1988). The advantage of this method is that it is easy to create tool paths in a plane. Disadvantages of this method include the computational burden and numerical instabilities, especially when the layouts to be projected are far away from the design surface.

Offset curve method: This method is applicable only the surface is represented in the parametric or algebraic forms. In this method, offsets of curves on the design surface are considered (Suresh and Yang, 1994; Maekawa, 1996). Since geodesic curves on general surfaces are analogous to straight lines on planes, the offsets are calculated along the geodesic curves normal to the progenitor curve. The advantage of this method is that one directly deals with distances along the surface. This method yields a constant scallop height on the manufactured surface. This method also yields the shortest overall length of tool paths compared to the previous tool path generation methods. However, geodesic curves are expensive to compute and offset curves on surfaces suffer from similar problems as those experienced on planes (such as cusps and self-intersections).

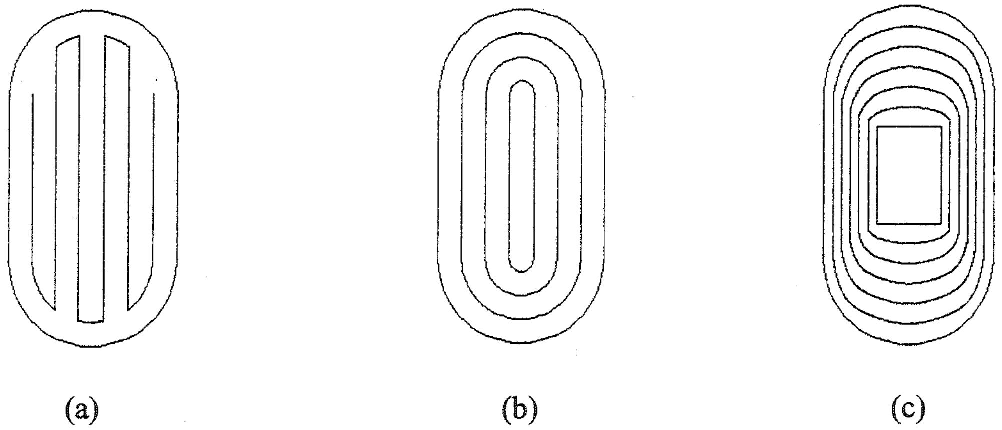
There are also some other tool path generation methods such as *piece curve method* (Austin et al., 1997; Shah et al., 1991), *Iso-curvature method* (Jensen and Anderson, 1992), etc.. Extensive reviews on tool path planning maybe found in Marshall and Griffiths (1994), Dragomatz and Mann (1997), Jenson and Anderson (1996), Sarma (2000).

1.2 2.5D NC Machining

The majority of industrial milling tasks can be performed using 2.5D machining. This is partially due to the fact that a surprisingly large number of mechanical parts are 2.5D and even the more complicated objects are usually produced from a billet by a 2.5D roughing and 3D finishing. Thus the computation of the tool path for 2.5D machining is one of the most important issues in NC machining.

Tool path planning is needed to execute both rough cuts and finish cuts in pocketing and contouring. Generation of the geometric profile of the tool path is the first stage. Interpolation methods are then applied to generate the NC code for machining.

Among tool path generation techniques for pocketing and contouring, *Zigzag* and *contour parallel* methods are widely used in practice.



(a) (b) (c)

Figure 1.1. Tool path generated from zigzag (a), contour parallel (b), and morphing (c).

In *zigzag* method as shown in Figure 1.1(a), the tools are moved along line segments which are parallel to a reference line selected initially. The zigzag tool paths are bi-directional tool paths which cause the tool to cut alternately along the spindle direction and then against it, giving respectively conventional and climb cuttings. This alternate change of cutting mode creates a difference in cutting speed, and leads to problems such as non-uniform surface quality, machine chatter and shorter tool life. A variant referred to as zig cutting has uni-directional tool paths. By using only one cutting direction the cutting speed is kept uniform.

However, there is a considerable amount of non-productive time involved in returning the tool to the start-cut position at the end of each tool path.

Contour-parallel cutting is shown in Figure 1.1(b). In this method, normally successive offsets of the original contours are first generated. These offsets are then chained together into a single spiraling tool path that follows the contours. To find the successive offsets of the boundary, pair-wise intersection may be used. Although this approach is complex in computation (requiring testing each pair of segments for possible intersection), it has been used extensively. Additionally, Voronoi diagrams can be used to determine the offset curves more efficiently (Persson, 1978; Held, Lukacs and Ando, 1994). Pixel based method (Choi and Kim, 1997) is another method for generating the offsets. But it is time-consuming and requires a large amount of memory because of the application of Z-map. The contour parallel method is generally favored, especially in the case of complex shape machining, because it gives less machining time, better surface quality in comparison with the zigzag method.

Morphing techniques shown in Figure 1.1(c) is another method to generate tool path for 2.5D machining. In the method, tool paths are generated by moving the cutting tool along a series of linearly interpolated paths between the outer boundary and island contour. Since the proportional blending offset creates uneven offset curves, and is difficult to implement for complex shape, the method is suitable for pocketing with only one island positioned approximately in the center of the pocket.

Offset curve generation is one of the key steps in contour parallel machining. Given a plane curve $r(t)$ with a regular parameterization on $t \in [0,1]$ the offset to $r(t)$ at a distance d is the curve defined by

$$r_o(t) = r(t) \pm dn(t) \qquad \text{For } t \in [0,1] \tag{1.1}$$

where $n(t)$ is the unit normal to $r(t)$ at each point. Offset curves are thus "parallel" to the original curve with constant normal distance. Depending on the shape of the generator curve, the offset curve can have loops that need to be trimmed out and gaps that need to be filled in for valid NC machining. For the most part, the geometrical features of the generator curve are retained by the offset curves.

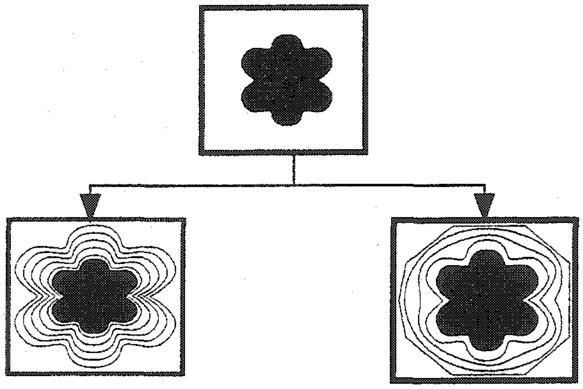
Once the tool path is generated, interpolation methods are used to generate NC code. In 2.5D machining, tool paths are approximated by straight lines or circular arc segments, as most CNC interpolators accommodate only such elements. Vickers and Bradley (1992) noted that the machining time is increased because of dwell between consecutive NC control commands. A large percentage of the machining time is spent either at less than optimum feed rate or actually waiting for the next instruction. For a given machining tolerance, a complex curve with more peaks and valleys needs more NC commands compared to a simpler (smooth) curve with less peaks and valleys for machining. Therefore, complex curves will require more machining time, compared to simpler curves of same length because of more dwell.

Machining time is a function of the length of the machining segment and the shape complexity of the machining segment for a given feedrate. As an example, consider machining of a complex 2D profile from rectangular stock. This is similar to pocket machining with an island except that the wall effects are ignored. If the conventional contour parallel technique is applied, the tool paths for rough cutting, which are far away from the object boundary, have the same shape and geometry complexity as the tool paths for fine cutting close to the object boundary. However, in the rough cutting stage there is no necessity for the tool path to have all the geometric details of the object boundary. The machining efficiency may be improved by using simpler curves far away from the object boundary.

In this research we demonstrate the application of multiresolution analysis based on the theory of wavelets in tool path generation. This is a novel application and provides added insight and a rich theoretical framework for some new tool path planning algorithms. The appearance of wavelets is a relatively recent development in mathematics. They give rise to hierarchical representation and have been successfully used for various tasks such as approximation theory, signal processing, image storage and compression, and content based retrieval. Recently the theory of wavelets and multiresolution analysis has also been applied to hierarchical editing of curves and surfaces in computer graphics.

The theory of wavelets allows the representation of curves and surfaces at various levels of detail. A complex curve may be represented at various scales. At the higher scales, the detail features on the curve are included. At the lower scales the detail features are smoothed out and an approximate curve that shows the general trend of the complex curve is obtained.

This hierarchical curve representation can be used for 2D contour machining. Figure 1.2 shows an illustration where, the region between the rectangular boundary and the shaded object needs to be removed. The tool paths are generated using the contour parallel method and alternatively using a hierarchical curve representation based on the B-spline wavelet. In the wavelet-based method, close to the 2D-object contour a high amount of detail is provide for accurate machining. However, as we move away from the 2D-object contour, the attention to detail is reduced. Low-resolution curves, after suitable offset modification, are used for the outer curves. Details are progressively added to the inner curves. This leads to an integrated rough and finish cutting strategy. The machining times may also be reduced because of the smoother curves in rough cutting. However, because of the variable offset that is introduced, there is potential for non-uniform chip thickness or uncut material to remain. The non-uniform chip thickness may be addressed by feedrate compensation to maintain a uniform cutting force if desired. To machine the uncut regions additional machining curves referred to as adaptive curves are used.



Contour parallel method

Multiresolution method

Figure 1.2. Cutting sequence for 2.5D object machining.

1.3 5-Axis NC Machining

To achieve the intended performance of a product (e.g., an impeller or turbine blade) or capture its desired aesthetics (e.g., a car fender or hood), engineers and designers continually seek new methods and techniques for specifying part geometry. This has resulted in the development and use of entirely new classes of surfaces.

Traditionally, the tool paths required to machine complex surfaces have been constructed using numerical control languages or CAD systems. These languages or systems generate numerical instructions for multi-axis machine tools based on defined part geometry and specified machining parameters. The numerical instructions are generated by first reducing the dimensionality of the surface to that of planar or isoparametric curves.

Milling machines used to machine complex surfaces have three axes of simultaneous tool positioning control and zero, one or two axes of simultaneous tool orientation control. While 3-axis milling machines (those with no tool orientation capabilities) have historically been the choice to machine complex surfaced parts, 5-axis milling machines have increasingly been used in industry. As compared to 3-axis machining, 5-axis machining (a simple example is shown in Figure 1.3) offers many advantages such as higher productivity and better machining quality. In 5-axis machining, the orientation of the tool can be determined by the two additional degrees of freedom so as to obtain efficient tool paths. Figure 1.4 shows the tool orientation control in 5-axis machining. The angle of the tool in the plane of motion is called the tool inclination angle α, and the angle of tool out of the plane of motion is called the tool tilt angle β.

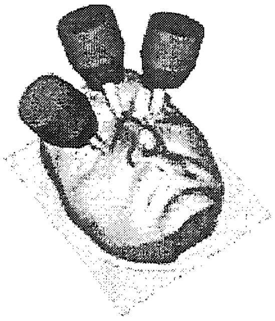


Figure 1.3. A 5-axis machining example

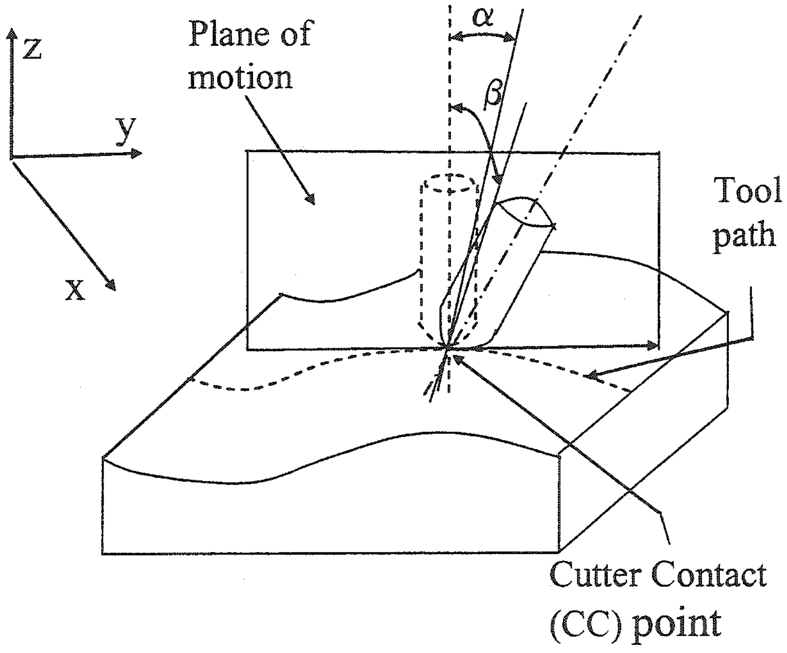


Figure 1.4. Tool orientation in 5-axis machining.

For complex shapes, 5-axis tool path generation is a difficult process. There are three functions that must be accomplished simultaneously: (1) selecting the right tool; (2) determining tool orientation; and (3) covering the surface. Tool paths for 5 axis NC machining are typically generated from CAD/CAM systems. However, currently available CAD/CAM systems generate tool paths for 5-axis machining assuming the user has picked the right tool, and for a fixed user provided tool orientation. Some advanced systems provide the ability to interpolate tool orientations based on fixed user provided tool orientations at sample points. Collision avoidance is not considered by the CAD/CAM systems and typically users turn to NC machining verification systems to visually identify regions of collision. If collisions are detected at this stage, the user must accordingly change the tool orientations and perhaps even the tool path. This becomes a trial and error process, and for complex shapes such as turbine blades, 5-axis tool path generation can be time-consuming and worker intensive.

Therefore, determining tool orientation is one of the most important steps in 5-axis machining. Research has focused on optimizing tool orientation with the intent of maximizing the material removal (Lee and Ji, 1997; Jensen et al., 1993; Kruth and Klewais, 1994; Redonnet et al., 1998; Rao et al., 1996) and for collision avoidance (Chen and Woo, 1992; Lee and Chang, 1995; Tseng and Joshi, 1994; Choi and Jerard 1998). Tool selection is another important problem in 5-axis NC machining. However, research on automatic tool selection for 5-axis machining is limited. Lee and Chang (1996) use the maximum effective cutting radius approach and Jenson et al. (2002) use a method based on curvature matched machining.

The above methods are computationally intensive involving surface curvature computation, visibility maps, or Voronoi diagrams, and are sometimes performed unnecessarily or are redundant because these decisions might be based on convex hull intersections. Moreover, current 5-axis tool path generation methods developed in the literature are not integrated with rough cut machining strategies and fine cut machining strategies being considered independently. The current research on tool path generation has been almost no consideration of generation of appropriate intermediate shapes as the references of NC programming. The tool cutting ability and removal volume are disregarded (when only one geometric model for a final product shape is used as a reference for all of the rough-to-fine machining operation), or only be performed by a very experienced production engineer (when a series of intermediate geometric models for each rough-to-fine machining operation is manually designed). Moreover, redesign of the models may be needed if the tools or product shape are modified. Further, in current tool path generation methods, different theoretical frameworks are used for several important analyses such as tool orientation, tool selection and (gouge avoidance).

Wavelets can provide a single theoretical framework for performing most of the geometric reasoning needed for efficient tool path generation in 5-axis NC machining. The application of wavelets can also minimize the need for these geometric algorithms and improve the computational efficiency when these geometric algorithms are needed. Most important, wavelet based multiresolution representations generate a series of intermediate shape models as the references of NC programming, in which the tool cutting ability and removal volume can be automatically considered, the amount of model data can be compressed, and the resolution of the models can be changed. In addition, multiresolution models represented by polygonal meshed surfaces are becoming more popular in CAD/CAM. However, there is almost no published literature about tool path generation directly from a multiresolution meshed representation. The wavelet representation is very conducive for tool path generation based on a meshed surface.

Wavelets and multiresolution analysis may be used to hierarchically represent the original surface. Each hierarchical model successively approximates the final object shape. Thus rough-cuts can be made with the low-resolution models and finish cuts can be made with the high-resolution models. Wavelets may also be used to develop new multiresolution accessibility analysis algorithms for efficient collision detection and avoidance.

The rest of the report is organized as follows. The theory of multiresolution analysis and wavelets is briefly reviewed in chapter 2. In chapter 3, the tool path for machining 2D profiles from rectangular stock, based on the endpoint interpolating B-spline wavelets is derived. In chapter 4, an improved multiresolution tool path based on the convex hull is presented. Tool path planning in 5-axis machining based on multiresolution analysis are proposed in chapter 5. Conclusions are presented in chapter 6.

CHAPTER 2 WAVELETS AND MULTIRESOLUTION ANALYSIS

This chapter provides some background on wavelets and multiresolution analysis. Multiresolution analysis is a simple mathematical tool that has found a wide variety of applications in recent years, including signal analysis (Mallet, 1989), image processing (DeVore *et al.*, 1992), and numerical analysis (Beylkin *et al.*, 1991). Informally, wavelets are the basis functions for multiresolution analysis.

A signal or a function may be better understood if expressed as a linear decomposition over a basis.

$$f(t) = \sum_{l} a_l \phi_l(t) \tag{2.1}$$

The basis functions are usually chosen to be orthogonal. For the Fourier series these basis functions are $\sin(k\omega_0 t)$ and $\cos(k\omega_0 t)$. On the other hand, the wavelet expansion is a two-parameter system described by

$$f(t) = \sum_{k} \sum_{j} a_{j,k} \psi_{j,k}(t) \tag{2.2}$$

The coefficients $a_{j,k}$ are called as the discrete wavelet transform. The wavelet expansion gives a time frequency localization of the signal. A wavelet representation is much like a musical score where the location of the notes tells when the tones occur and what their frequencies are. Wavelet systems are generated from a single scaling function by simple scaling and translation. Therefore, if a set of functions is represented by a weighted-sum of $\psi(t-k)$ then a larger set of functions may be represented by $\psi(2t-k)$. Wavelets thus satisfy the multiresolution conditions. The lower resolution coefficients can be calculated from the higher resolution coefficients, using a tree structured algorithm called a *filter bank.*

Consider a discrete signal C^n, expressed as a column vector of samples $\left[c_1^n, ..., c_m^n\right]^T$. In our curve application described in Chapter 3, for example, the samples c_i^n will be the curve's control points in $\Re^2$.

Suppose we wish to create a low-resolution version C^{n-1} of C^n with a fewer number of samples $m^{'}$. The standard approach for creating the $m^{'}$ samples of C^{n-1} is to use some form of linear filtering and subsampling on the *m* samples of C^n. This process can be expressed as a matrix equation

$$C^{n-1} = A^n C^n \tag{2.3}$$

Where A^n is an $m^{'}$ x m matrix.

Since C^{n-1} contains fewer samples than C^n, it is intuitively clear that some amount of details is lost in this filtering process. If A^n is chosen appropriately, it is possible to capture the lost detail as another signal D^{n-1} with m - $m^{'}$ samples, computed by

$$D^{n-1} = B^n C^n \tag{2.4}$$

Where B^n is an $(m$ - $m^{'})$ x m matrix, which is related to matrix A^n. The pair of matrices A^n and B^n are called *analysis filters*. The process of splitting a signal C^n into a low-resolution version C^{n-1} and detail D^{n-1} is called *decomposition.* Usually m' is roughly half of m, so that C^{n-1} and D^{n-1} are roughly equal in size.

Note that C^{n-1} and D^{n-1} together have the same amount of information as C^n. If A^n and B^n are chosen correctly, then the original signal C^n can be recovered from C^{n-1} and D^{n-1} by using another pair of matrices P^n and Q^n, called *synthesis filters*, as follows:

$$C^n = P^n C^{n-1} + Q^n D^{n-1} \tag{2.5}$$

Recovering C^n from C^{n-1} and D^{n-1} is called *reconstruction.*

The procedure for splitting C^n into a low-resolution part C^{n-1} and a detail part D^{n-1} can be applied recursively to the new signal C^{n-1}. Thus, the original signal can be expressed as a hierarchy of low-resolution signals $C^0, \ldots, C^{n-1}$ and details $D^0, \ldots, D^{n-1}$, as shown in Figure 2.1. This recursive process is known as a *filter bank.*

$$\begin{tikzcd}
C^n \arrow[r, "A^n"] \arrow[dr, "B^n"'] & C^{n-1} \arrow[r, "A^{n-1}"] \arrow[dr, "B^{n-1}"'] & C^{n-2} & \cdots & {} \arrow[r, "A^1"] \arrow[dr, "B^1"'] & C^0 \\
& D^{n-1} & D^{n-2} & \cdots & & D^0
\end{tikzcd}$$

Figure 2.1. The filter bank

Since the original signal C^n can be recovered from the sequence $C^0, D^0, D^1, \ldots, D^{n-1}$, this sequence can be thought of as a transform of the original signal signal known as a *wavelet transform*. Note that the total size of the transform $C^0, D^0, \ldots, D^{n-1}$ is the same as that of the original signal C^n, so no extra storage is required.

In the following paragraphs we provide a simple introduction to wavelets and multiresolution analysis using Haar wavelets. The B-spline wavelets and the second generation wavelets are discussed next.

2.1 Haar Wavelets

To simply illustrate wavelets consider a sequence of numbers for instance with values

$$8, \quad 6, \quad 3, \quad 5$$

This sequence can be represented using the Haar basis wavelet transform. First, the numbers are averaged in a pair wise manner. This results in the sequence

$$7, \quad 4$$

This sequence however has some missing information from the initial sequence. More information can be added using "detail coefficients". The first detail coefficient may be taken to be 1 and the second detail coefficient as –1. We can recover the original numbers by the operations

$$\begin{aligned}7+1&=8\\7-1&=6\\4+(-1)&=3\\4-(-1)&=5\end{aligned}$$

Thus the initial sequence (8 6 3 5) can be represented as (7 4 1 –1). The numbers (7 4) are the coarse values and (1 –1) are the detail values. This can be carried one step further to yield (5.5,1.5,1,-1) and is the wavelet transform of (8,6,3,5).

For piece-wise continous functions, consider the open interval [0, 1). A one number sequence is just a function that is constant over the interval [0, 1). This may be represented as V^0 . A two number sequence has two constant pieces over [0 ½) and [1/2 1). This *space* is denoted as V^1. If this is continued further, V^j will include all piece-wise constant functions in the interval [0 1) with constant pieces over each of 2^j intervals. It is noted that the spaces V^j are nested and permits multiresolution analysis.

$$V^0 \subset V^1 \subset V^2 \subset$$

The basis functions for V^j are called scaling functions. A simple representation for V^j is the set of scaled and translated "box" functions

$$\phi_i^j(x) = \phi(2^j x - i), \quad i = 0,.....2^j - 1$$

where, (2.6)

$$\phi(x) = \begin{cases} 1 & \text{for } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases}$$

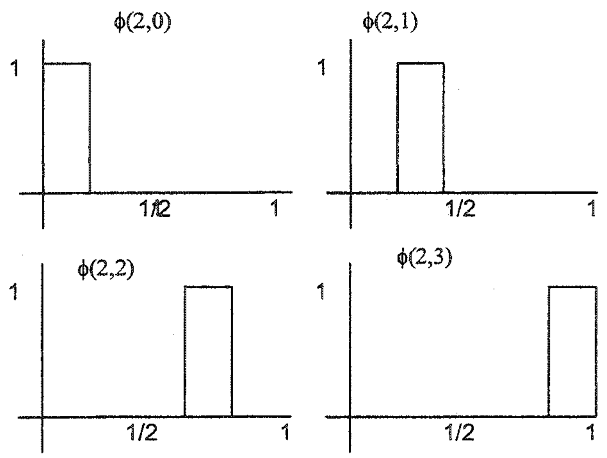


Figure 2.2. The Box basis for V^2.

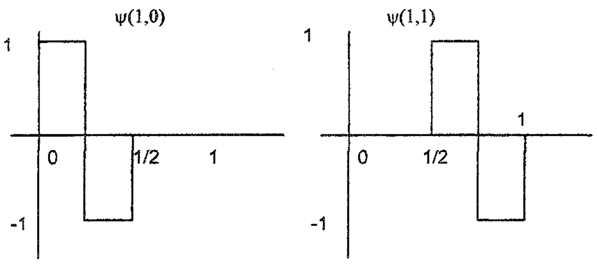


Figure 2.3. The Haar wavelets for W^1.

Figure 2.2 shows the four box functions for V^2. A new vector space defined as W^j is chosen as the orthogonal component of V^j in V^{j+1}. We can think of W^j as containing the detail in V^{j+1} that cannot be represented in V^j. A collection of linearly independent functions spanning W^j are called wavelets. The wavelets corresponding to the box basis are known as Haar wavelets. The Haar wavelets for W^1 are as in Figure 2.3 and can be specified as

$$\psi_i^j(x) = \psi(2^j x - i) \quad i = 0,\ldots 2^{j-1}$$

where,

$$\psi(x) = \begin{cases} 1 & \text{for } 0 \le x \le 1/2 \\ -1 & \text{for } 1/2 \le x \le 1 \end{cases} \tag{2.7}$$

$$\psi(x) = 0 \text{ otherwise}$$

The sequence (8,6,3,5) can now be represented in V^2 as

$$8\phi_0^2(x) + 6\phi_1^2(x) + 3\phi_2^2(x) + 5\phi_3^2(x)$$

It can also be expressed in V^1+W^1 as

$$7\phi_0^1(x) + 4\phi_1^1(x) + 1\psi_0^1(x) - 1\psi_1^1$$

Finally it can be expressed in V^0+W^2 as

$$5.5\phi_0^0(x) + 1.5\psi_0^0(x) + 1\psi_0^1(x) - 1\psi_1^1(x).$$

2.2 B-Spline Wavelets

While the Haar basis functions offer advantages in terms of simplicity, orthogonality and very small support, they suffer from a lack of continuity. To this end, B-spline wavelets were developed by Chui (1992). This is a class of wavelets with *k* continuous derivatives constructed from piecewise-polynomial splines. In fact the Haar basis is the simplest instance of spline wavelets, resulting when the polynomial degree is set to zero.

In particular, the cubic endpoint-interpolating B-spline functions defined on a closed interval are of interest. The endpoint interpolating B-spline wavelet is one of the important wavelets used in hierarchical representation of curves and surfaces, and allows the decomposition and reconstruction of multiresolution shape functions using matrix calculations.

To construct the endpoint-interpolating B-spline wavelets, three steps are needed.

1. Define the scaling functions for a nested set of function spaces,
2. Define an inner product, and
3. Obtain the wavelet functions.

The following paragraphs give a brief description on the generation of the cubic endpoint interpolating B-spline wavelet.

Step 1:

Given a positive integer $k > 3$, and a set of non-decreasing values $x_0, \ldots, x_{k+4}$ called *knots*, the nonuniform B-spline basis functions of degree 3 are defined recursively as follows.

For *i*=0, k, and for *r* =1, 2, 3, let

$$N_i^0(x) := \begin{cases} 1 & \text{if } x_i \le x < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

$$N_i^r(x) := \tfrac{x - x_i}{x_{i+r} - x_i} N_i^{r-1}(x) + \tfrac{x_{i+r+1} - x}{x_{i+r+1} - x_{i+1}} N_{i+1}^{r-1}(x).$$

(Note: The fractions in these equations are taken to be 0 when their denominators are 0.)

The endpoint-interpolating B-splines of degree 3 on the interval [0, 1] are obtained when the first and last 4 knots are set to 0 and 1, respectively. In this case, the functions $N_0^3(x),...,N_k^3(x)$ form a basis for the space of piecewise-polynomials of degree 3 with 2 continuous derivatives. For uniformly spaced cubic B-splines, $k = 2^j + 2$ and $x_4,...,x_k$ are chosen to produce 2^j equally spaced interior intervals. This construction gives $2^j + 3$ B-spline basis functions for degree 3 and level *j*, and form the endpoint interpolating cubic B-spline scaling functions. The knot vector used to define the function is:

$$0, 0, 0, 0, \tfrac{1}{2^j}, \tfrac{2}{2^j}, ..., 1-\tfrac{1}{2^j}, \ 1, 1, 1, 1$$

At any hierarchical level *j* there are $2^j + 3$ control points and $2^j + 7$ knots. Figure 2.4 shows a hierarchical representation of a curve using B-spline wavelets. Figure 2.4(a) is the original curve with 11 control points at scale 3. Figure 2.4(b) is the curve at scale 2 with 7 control points. Figure 2.4(c) shows the curve at scale 1 with 5 control points. The curve shown at Figure 2.4(d) is at scale 0 (lowest scale) with 4 control points.

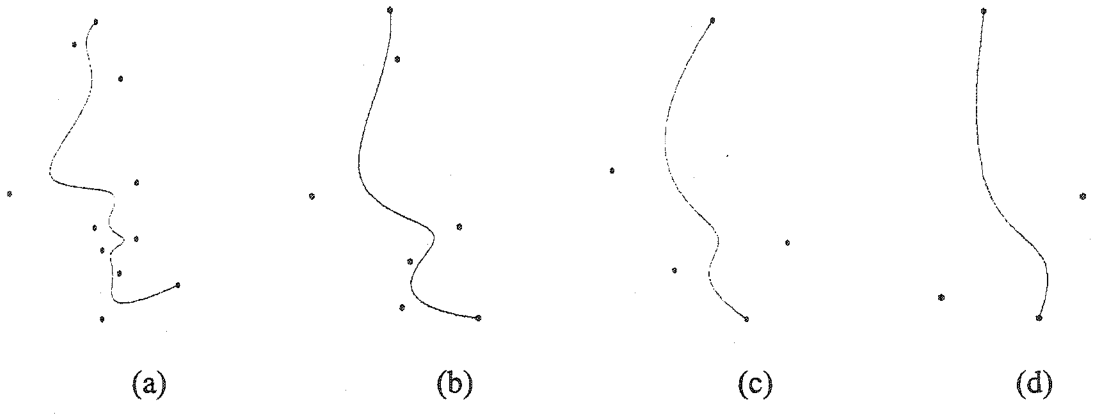


Figure 2.4. Hierarchical representations of a curve based on B-spline wavelets.

The condition that the function space V^j be nested is equivalent to requiring that the scaling functions be refinable. That is, for all j = 1, 2, … there must exist a matrix of constants P^j (this is the synthesis filter described in Equation 2.5) such that

$$\Phi^{j-1}(x) = \Phi^j(x)P^j \tag{2.9}$$

Where, Φ^j is a row matrix of all scaling functions at level j as shown in Equation 2.10.

$$\Phi^j = \left[\phi_0^j \quad \phi_1^j \quad \cdots, \quad \phi_{2^j+2}^j\right] \tag{2.10}$$

The synthesis filter P^j is of dimension (2^j+3) x $(2^{j-1}+3)$. The entries of the synthesis filter can be developed using the theory of B-splines (Chui and Quak, 1992; Quak and Weyrich, 1994). In Figure 2.5, matrix P^j is shown. The middle columns, for $j \geq 3$, are given by vertical translates of the fourth column, shifted down by 2 places for each column.

$$P^1 = \frac{1}{16}\begin{bmatrix} 16 & 0 & 0 & 0 \\ 8 & 8 & 0 & 0 \\ 0 & 8 & 8 & 0 \\ 0 & 0 & 8 & 8 \\ 0 & 0 & 0 & 16 \end{bmatrix}$$

$$P^2 = \frac{1}{16}\begin{bmatrix} 16 & 0 & 0 & 0 & 0 \\ 8 & 8 & 0 & 0 & 0 \\ 0 & 12 & 4 & 0 & 0 \\ 0 & 3 & 10 & 3 & 0 \\ 0 & 0 & 4 & 12 & 0 \\ 0 & 0 & 0 & 8 & 8 \\ 0 & 0 & 0 & 0 & 16 \end{bmatrix}$$

Figure 2.5. The refinement matrix P^j for cubic B-splines.

$$P^{j\geq 3} = \frac{1}{16}\begin{bmatrix} 16 & 0 & 0 & 0 & 0 & 0 & \\ 8 & 8 & 0 & 0 & 0 & 0 & \\ 0 & 12 & 4 & 0 & 0 & 0 & \\ 0 & 3 & 11 & 2 & 0 & 0 & \\ 0 & 0 & 8 & 8 & 0 & 0 & \cdots \\ 0 & 0 & 2 & 12 & 2 & 0 & \\ 0 & 0 & 0 & 8 & 8 & 0 & \\ 0 & 0 & 0 & 2 & 12 & 2 & \\ 0 & 0 & 0 & 0 & 8 & 8 & \\ 0 & 0 & 0 & 0 & 2 & 12 & \\ & & \vdots & & & & \ddots \end{bmatrix}$$

Figure 2.5. Continued

Step 2:

The second step is the choice of an inner product and the standard inner product is used for this purpose (Equation 2.11)

$$\langle f \mid g \rangle := \int_0^1 f(x)g(x)dx \tag{2.11}$$

Step3:

The final step is to find basis functions for the spaces W^j that are orthogonal complements to the space V^j.

Since the wavelets space W^{j-1} is by definition also a subspace of V^j, the wavelets $\Psi^{j-1}(x)$ can be written as linear combinations of the scaling functions $\Phi^j(x)$. This means:

$$\Psi^{j-1}(x) = \Phi^{j}(x)Q^{j} \tag{2.12}$$

Q^{j} is a $(2^{j}+3)\mathrm{x}(2^{j-1})$ matrix (This is the synthesis analysis described in Equation 2.5), where Ψ^{j-1} is a row matrix of wavelet functions as in Equation 2.13.

$$\Psi^{j-1} = \left[\psi_{0}^{j-1} \quad \psi_{1}^{j-1} \quad \cdots, \quad \psi_{2^{j-1}-1}^{j-1}\right] \tag{2.13}$$

Since all functions in $\Phi^{j-1}(x)$ must be orthogonal to all functions in $\Psi^{j-1}(x)$, we know that $\left\langle \phi_{k}^{j-1} \mid \psi_{l}^{j-1} \right\rangle = 0$ for all *k* and *l*. In order to deal with all these inner products simultaneously, the notation $[\left\langle \Phi^{j-1} \mid \Psi^{j-1} \right\rangle]$ is used to denote the matrix whose (*k*, *l*) entry is $\left\langle \phi_{k}^{j-1} \mid \psi_{l}^{j-1} \right\rangle$. The orthogonality conditions on the wavelet can be rewritten as:

$$[\left\langle \Phi^{j-1} \mid \Psi^{j-1} \right\rangle] = 0 \tag{2.14}$$

Substituting Equation 2.12 into Equation 2.14 yields

$$[\left\langle \Phi^{j-1} \mid \Phi^{j} \right\rangle]Q^{j} = 0 \tag{2.15}$$

By imposing some additional conditions, such as small support, to the homogeneous system of linear equations, the matrix Q^{j} and the corresponding B-spline wavelets can be obtained (shown in Figure 2.6).

$$Q^1 = \frac{1}{3}\begin{bmatrix} 1 \\ -2 \\ 3 \\ -2 \\ 1 \end{bmatrix}$$

$$Q^2 = \frac{1}{2064}\begin{bmatrix} -1368 & 0 \\ 2064 & 240 \\ -1793 & -691 \\ 1053 & 1053 \\ -691 & -1793 \\ 240 & 2064 \\ 0 & -1368 \end{bmatrix}$$

$$Q^3 = \begin{bmatrix}
\frac{-394762}{574765} & 0 & 0 & 0 \\
1 & \frac{-7166160}{28124263} & 0 & 0 \\
\frac{-33030599}{41383080} & \frac{333497715}{478112471} & \frac{6908335}{478112471} & 0 \\
\frac{633094403}{1655323200} & \frac{-881412943}{956224942} & \frac{-74736797}{956224942} & \frac{27877}{1655323200} \\
\frac{-19083341}{137943600} & 1 & \frac{8833647}{28124263} & \frac{-864187}{413830800} \\
\frac{4681957}{165532320} & \frac{-689203555}{956224942} & \frac{-689203555}{956224942} & \frac{4681957}{165532320} \\
\frac{-864187}{413830800} & \frac{8833647}{28124263} & 1 & \frac{-19083341}{137943600} \\
\frac{27877}{1655323200} & \frac{-74736797}{956224942} & \frac{-881412943}{956224942} & \frac{633094403}{1655323200} \\
0 & \frac{6908335}{478112471} & \frac{333497715}{478112471} & \frac{-33030599}{41383080} \\
0 & 0 & \frac{-7166160}{28124263} & 1 \\
0 & 0 & 0 & \frac{-394762}{574765}
\end{bmatrix}$$

Figure 2.6. The matrix Q^j for Cubic B-splines.

$$Q^{j\geq 4} = \begin{bmatrix} \frac{-394762}{574765} & 0 & 0 & 0 & \\ 1 & \frac{-1050072320}{4096633377} & 0 & 0 & \\ \frac{-33030599}{41383080} & \frac{2096854390}{2989435167} & \frac{307090}{19335989} & 0 & \\ \frac{633094403}{1655323200} & \frac{-11070246427}{11957740668} & \frac{-6643465}{77343956} & \frac{-1}{24264} & \\ \frac{-19083341}{137943600} & 1 & \frac{6646005}{19335989} & \frac{31}{6066} & \\ \frac{4681957}{165532320} & \frac{-157389496903}{221218202358} & \frac{-29839177}{38671978} & \frac{-559}{8088} & \\ \frac{-864187}{413830800} & \frac{1732435193}{5821531641} & 1 & \frac{988}{3033} & \cdots \\ \frac{27877}{1655323200} & \frac{-27809640281}{44243640471 6} & \frac{-58651607}{77343956} & \frac{-9241}{12132} & \\ 0 & \frac{171326708}{36869700393} & \frac{6261828}{19335989} & 1 & \\ 0 & \frac{-1381667}{36869700393} & \frac{-1328199}{19335989} & \frac{-9241}{12132} & \\ 0 & 0 & \frac{98208}{19335989} & \frac{988}{3033} & \\ 0 & 0 & \frac{-792}{19335989} & \frac{-559}{8088} & \\ 0 & 0 & 0 & \frac{31}{6066} & \\ 0 & 0 & 0 & \frac{-1}{24264} & \\ & \vdots & & & \ddots \end{bmatrix}$$

Figure 2.6. Continued

The scaling functions and the wavelet functions have now been found. To use B-splines wavelets, *wavelet decomposition* (Equation 2.16) and *wavelet reconstruction* (Equation 2.17) need to be implemented.

$$[P^j \mid Q^j][\tfrac{c^{j-1}}{d^{j-1}}] = c^j \tag{2.16}$$

$$c^j = P^j c^{j-1} + Q^j d^{j-1} \tag{2.17}$$

Where, c^j and d^j are column matrices of the corresponding coefficients. The coefficient matrix c^j can be thought of as the *x* and *y*-coordinates of a curve's control points in $\Re^2$. The wavelet decomposition allows one to decompose a curve into a lower or coarse scale. As shown in Equation 2.16 the scaling coefficients c^{j-1} and wavelet coefficients d^{j-1} are obtained by solving the linear system. The wavelet reconstruction is used to recover the original or higher level curve from the lower level curve as in Equation 2.17. The wavelet coefficients may be changed to edit the curve details and the scaling coefficients may be changed to edit the sweep of the curve. A detailed description of applying B-spline wavelets to curve or surface editing may be found in (Finkelstein and Salesin, 1994; Stollnitz *et al.*, 1995).

2.3 Second Generation Wavelets

2.3.1 Basic Ideas

Classic wavelet constructions define the wavelet functions as dyadic translates and dilates of one particular, fixed function. These traditional wavelets are referred *as first generation wavelet*. They have been extensively used for representing, manipulating, or compressing data on the real line or plane but they are not sufficient when it comes to multiresolution analysis of meshes with arbitrary topology that are most frequently found in computer graphics applications. It has been shown that all finite classical wavelets may be represented using *second generation wavelets*.

The main philosophy behind second generation wavelets is to achieve all desirable properties for wavelets adapted to more general settings than the real line or plane. The key realization here is the fact that translation and dilation are not required in order to obtain wavelets with the desired properties. The only invariant that needs to be maintained is that a basis function can be expressed as a finite linear combination of wavelet basis functions in a deeper, finer level.

First generation wavelets are typically constructed with Fourier analysis because translation and dilation become algebraic operations under the Fourier transform. In the case of second generation wavelets, though, the Fourier transform can no longer be used as a construction tool. An alternative construction method, called the *lifting scheme* was introduced by Sweldens (1995).

Lifting allows us to build our bases in a fully biorthogonal framework. This ensures that all bases are of finite (and small) support and the resulting filters are small and easy to derive. The filterbank algorithm of the second generation wavelet is briefly presented as follows. For more detailed theory about the lifting scheme and second generation wavelets, readers can refer to Sweldens (1995) and Schroder and Sweldens (1995).

2.3.2 Filterbank Algorithm

Given a base mesh / final mesh pair (M^0, M^n respectively), the construction generates n hierarchical levels of resolution such that

$$M^0 \subset M^1 \subset \cdots \subset M^j \subset \cdots \subset M^{n-1} \subset M^n.$$

We define $\phi_i^j(x)$ to be the *i*th scaling function at resolution *j*, while *x* represents a point over the domain. $\Phi^j(x)$ is defined as the matrix consisting of the *i* functions $\phi_i^j(x)$.

We now write the matrix $\Phi^j(x)$ as

$$\Phi^j(x) = [O^j(x) \quad N^j(x)] \tag{2.18}$$

Where $O^j(x)$ consists of the scaling functions $\phi_i^{j+1}(x)$ associated with the old vertices of M^j, while $N^j(x)$ refers to the scaling functions associated with vertices added to the last mesh.

Let $\psi_i^j(x)$ denote the *i*th locally supported wavelet approximation, and let $\Psi^j(x)$ be the row matrix of these functions. The analysis and synthesis filters are defined by

$$[\Phi^j(x) \quad \Psi^j(x)] = \Phi^{j+1}(x)[P^j \quad Q^j] \tag{2.19}$$

and

$$\begin{bmatrix} A^j \\ B^j \end{bmatrix} = [P^j \quad Q^j]^{-1} \tag{2.20}$$

respectively.

The lifting scheme is used to construct biorthogonal surface wavelets. The strategy employed is to construct "lazy wavelets" $\Psi_{lazy}^{j-1}(x)$ consisting of the scaling functions associated with the midpoints of the edges of M^{j-1}.

For lazy wavelet construction, we define

$$[P_{lazy}^{j} \quad Q_{lazy}^{j}] = \left[\begin{array}{cc} \begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{matrix} & 0 \\ P_{m}^{j} & \begin{matrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{matrix} \end{array}\right] \tag{2.21}$$

Where P_{m}^{j} is merely the matrix of connectivity information of the new vertices at step j. In order to construct "k-disk" wavelets (Stollnitz, Derose and Salesin, 1996) the above matrices are modified by the matrix S^{j} (defined in Equation 2.24) in the following fashion:

$$[P_{kd}^{j} \mid Q_{kd}^{j}] = [P_{lazy}^{j} \mid Q_{lazy}^{j} - P_{lazy}^{j} S^{j}] \tag{2.22}$$

$$\left[\frac{A_{kd}^{j}}{B_{kd}^{j}}\right] = \left[\frac{A_{lazy}^{j} + S^{j} B_{lazy}^{j}}{B_{lazy}^{j}}\right] \tag{2.23}$$

$$[< \Phi^{j} \mid \Phi^{j} >] S^{j} = (P^{j})^{T} [< \Phi^{j+1} \mid N^{j+1} >] \tag{2.24}$$

Let V^{j} denote the column vector of vertices of M^{j}, and W^{j} denote the corresponding matrix of wavelet coefficients. Analysis can be defined by

$$V^j = A^j V^{j+1} \tag{2.25}$$

$$W^j = B^j V^{j+1} \tag{2.26}$$

while synthesis is defined by

$$V^{j+1} = P^j V^j + Q^j W^j \tag{2.27}$$

2.3.3 Subdivision Connectivity of Arbitrary Meshes

Lounsbery (1995) pioneered the multiresolution representation of a mesh using subdivision wavelets, later generalized to second generation wavelets by Sweldens, Schroder and others. The multiresolution representation of a mesh consists of a simple base mesh together with a sequence of local correction terms, called wavelet coefficients, capturing the detail present in the object at various resolutions.

These methods have a prerequisite that the input mesh have subdivision connectivity, or in other words, that the mesh can be obtained from a simple base mesh by recursive subdivision (typically, 4-to-1 triangle splitting). Such a mesh with subdivision connectivity is referred to as a semi regular mesh. The complex meshes obtained from automatic laser range scanning and digitizing, for example, certainly do not have the subdivision connectivity property, which is a serious limitation. Voronoi tiling (Eck *et al.*, 1995), fine to coarse decimation

(Lee *et al.*, 1998) and volume data sets (Wood *et al.*, 2000) are strategies that have been used to generate a semi-regular mesh from an arbitrary mesh.

CHAPTER 3 TOOL PATH PLANNING FOR 2.5D MACHINING

In this chapter, a method to generate NC tool path curves using endpoint–interpolating B-spline wavelets is developed. The tool path consists of basic tool paths and adaptive tool paths. A complex curve is divided into smooth and non-smooth segments using corner point detection based on curvature analysis. The smooth segments are offset by moving the control nets. The non-smooth segments are first decomposed into a coarse scale and then the coarse scale control nets are offset. Closer to the object a higher scale is used for the decomposition to ensure that the contour of the object is followed closely. Farther away from the object a lower scale is used, as there is no necessity to follow the object contour closely. Adaptive curves are used to fill in between adjacent offset curves in non-smooth regions. This strategy leads to an integrated rough and finish cut method for tool path generation.

The generation of tool paths for 2.5D objects based on wavelets broadly involves the following steps.

1. Detect corner points in the initial contour based on curvature analysis;
2. Partition the whole contour into smooth regions (low density) and non-smooth (high density) regions;
3. Wavelet decompose non-smooth regions to adjacent lower level;
4. Offset the contour based on wavelets. Smooth regions and non-smooth regions are offset with different distances;

5. Repeat steps 3 and 4 to current offset curve to generate the machining path sequence progressively outwards;
6. Add adaptive path curves between adjacent offset curves in non-smooth regions.

These steps are also shown in Figure 3.1.

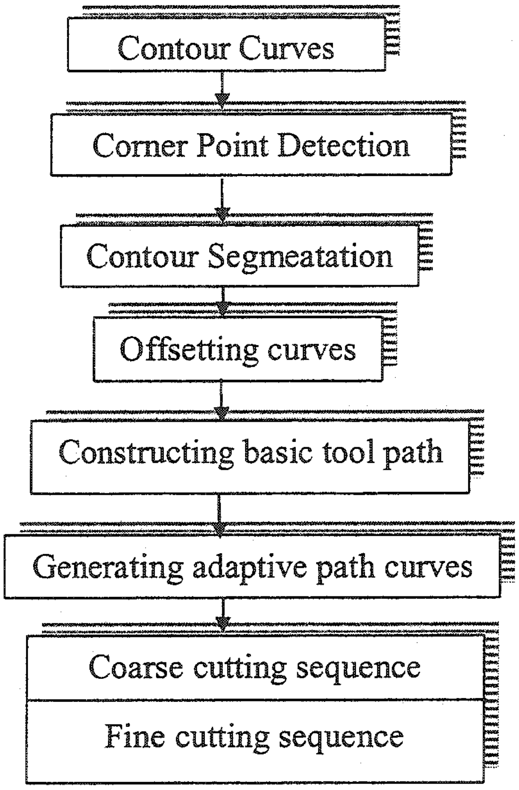


Figure 3.1. Schematic of steps for wavelets based multiresolution tool path generation.

The overall algorithm for wavelet based tool path generation is shown in Figure 3.2.

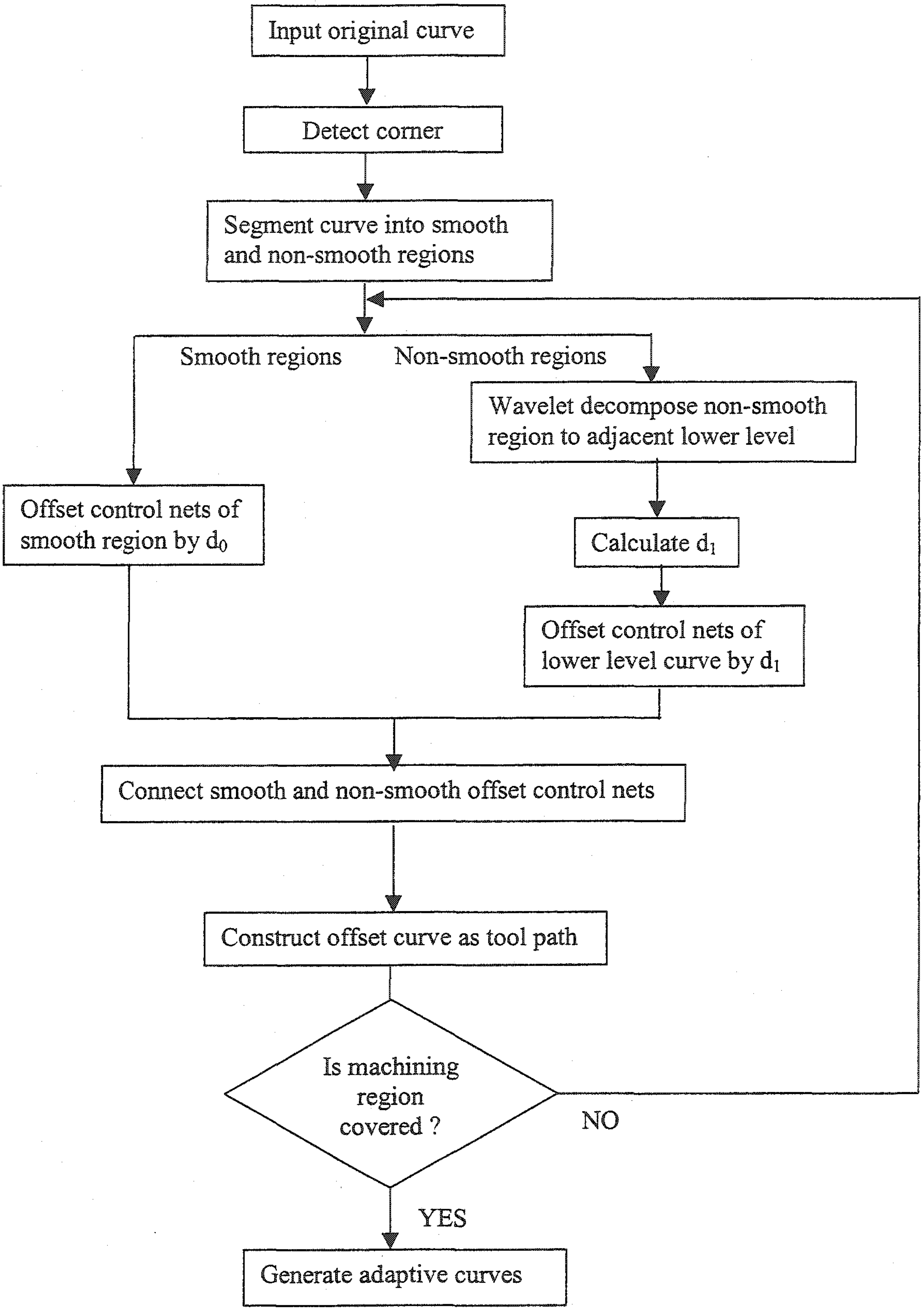

Input original curve
Detect corner
Segment curve into smooth and non-smooth regions
Smooth regions
Non-smooth regions
Wavelet decompose non-smooth region to adjacent lower level
Offset control nets of smooth region by d_0
Calculate d_1
Offset control nets of lower level curve by d_1
Connect smooth and non-smooth offset control nets
Construct offset curve as tool path
Is machining region covered ?
NO
YES
Generate adaptive curves

Figure 3.2. Flowchart for multiresolution tool path curve generation

3.1 Corner Point Detection

A corner point is a dominant point in a curve where either the gradient of the curvature is very steep or there is a curvature discontinuity. We adopt the algorithm developed by Wang *et al.* (1999) to detect the corner points. Curvature functions of a B-spline curve at different scales j can be estimated using

$$k(u,j) = \frac{W_1x(u,j)W_2y(u,j) - W_1y(u,j)W_2x(u,j)}{(W_1x(u,j)^2 + W_1y(u,j)^2)^{3/2}} \tag{25}$$

Where, u is the arc length. The B-spline wavelet transforms $W_1C(u,j)$ and $W_2C(u,j)$ of the curve $C(u,j)$ are defined as the convolution of the curve with the first and second derivatives of the nth-order B-spline, respectively. Wavelet transform vectors $[W_1x(u,j), W_1y(u,j)]^T$ and $[W_2x(u,j), W_2y(u,j)]^T$ are computed using a fast subdivision scheme.

The corner points of the curve correspond to the locations of peaks in the above curvature functions. The multiscale information is used to trace these corner points and correct their locations with a coarse-to-fine matching strategy. Among the candidate corner points obtained from the above procedure, those with significant changes in direction are identified as corner points.

3.2 Object Contour Segmentation

After the corner points are detected, the object contour is segmented into smooth regions and non-smooth regions based on the density of corner points. Non-smooth regions are curve sections where a large number of corner-points are concentrated in a comparatively small section of the curve. Geometrically speaking these are curve segments in which peaks and valleys appear frequently. A smooth region of the curve is the section with few corner points or without any corner point.

As a simple rule that works well for test cases that we have experimented with, the object contour is segmented based on a threshold value $v = l/m$, where l is the length of the object contour and m is the number of corner points. The parameter v gives the average length between two corner points. If the arc length between two adjacent corner points is larger than the threshold value v, then this section is intuitively a smooth section for the object contour under consideration. First, smooth sections are identified based on this threshold value. Next, the segment of the object contour between two adjacent smooth regions is identified as a non-smooth region. The entire object contour is thus decomposed into smooth and non-smooth regions.

An example of segmentation is shown in Figure 3.3. Points A, B, C, D, E, F, G and H are corner points detected in the curve. Segments AH and DE are identified as smooth regions. Segments ABCD and EFGH are the non-smooth regions.

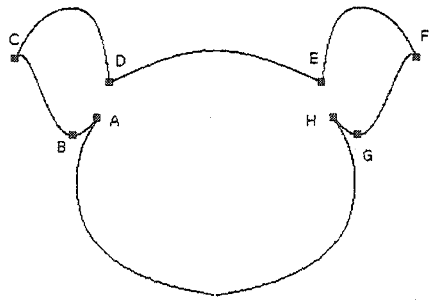
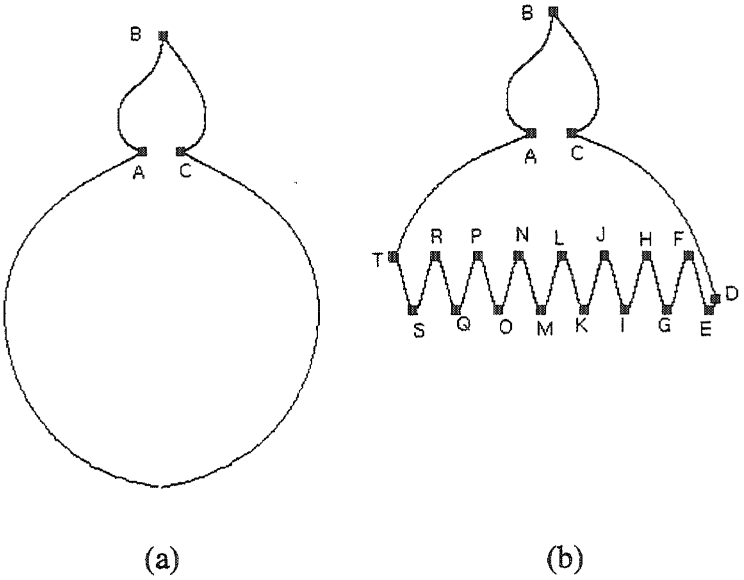


Figure 3.3. Segmentation of object contour based on corner point detection.



(a)

(b)

Figure 3.4. Relative smooth and non-smooth regions on an object contour.

In Figure 3.4 we compare two objects with a similar feature. In Figure 3.4(a) segment ABC is identified as a non-smooth region. However, in Figure 3.4(b) the segment ABC is identified as a smooth region. Therefore, this simple threshold based segmentation identifies the principal non-smooth regions in an object, for example arc segment (D, …, T) in Figure

3.4(b). We believe that this type of object segmentation is reasonable, as the wavelet decomposition is most useful in the principal non-smooth regions.

3.3 Wavelet Decomposition

Each non-smooth region at scale level *j* is decomposed to the adjacent lower level (*j*-1) using wavelet decomposition (Equation 2.16). The new curve segments in non-smooth regions can be considered as an approximation to the original segments. The wavelet coefficients found in this step are used later in selecting the offset distance in non-smooth regions. The smooth regions are left unchanged.

3.4 Basic Offset Curve Generation

Offsetting the object contour is a necessary procedure to generate tool paths. In this research, the offset curve is obtained by offsetting the control polygon after wavelet decomposition. The two algorithms, namely *Tool path curve* and *Offset* are used to generate a single tool path curve. The object contour is assumed to comprise of smooth segments $S_1, S_2, \ldots, S_u$ and non-smooth segments $N_1, N_2, \ldots, N_t$. The algorithm *Tool path curve* decomposes each non-smooth region into the adjacent lower level. In the decomposition process, if the higher level curve does not have the number of control points as required for multiresolution analysis, then additional control points are inserted to match the scale requirements. The smooth regions $S_1, S_2, \ldots, S_u$ are left unchanged. The algorithm *Offset* finds the offset curve to the smooth and non-smooth regions based on Tiller and Hanson's method (1984), which compares well with other offset techniques (Elber *et al.*, 1997).

```
**Algorithm** *Tool path curve*
**Begin**
    $i = 1$;
    **While** $i \leq u$ (number of smooth regions) **do**
        Offset the control net of smooth region $S_i$ by a distance $d_0$ (Refer to **Algorithm** *offset*);
        $i = i + 1$;
    **End;**
    $i = 1$;
    **While** $i \leq t$ (number of non-smooth regions) **do**
        Identify $n_{cp}$ = number of control points in non-smooth region N$_i$;
        Scale level $j = \log_2(n_{cp} - 3)$;
        **While** $j$ != integer **do**
            Insert an additional control point to $N_i$ by subdivision;
            Recalculate $j$;
        **End;**
        Wavelet decompose $N_i$ to lower scale level $j - 1$ to form wavelet curve $N_i^{j-1}$;
        Offset the control net of N$_i^{j-1}$ by a distance $d_1$ (Refer to **Algorithm** *offset*);
        $i = i + 1$;
        End;
        Connect the offset control nets of both smooth regions and non-smooth regions in the original order;
        Construct B-spline offset curve O from the connected control net;
        Return curve O;
    End;
```

```
    Algorithm Offset
    Begin
        Input a smooth region S or non-smooth region N;
        Identify the control net u, the set of control points;
        Deviation := $\infty$;
        Specify offset tolerance $\varepsilon$;
        While deviation > $\varepsilon$ do
            Offset normally each leg of u (as line segments) a distance d ($d = d_0$ for smooth region; $d = d_1$ for non-smooth region) to form a new set of control points v;
            Construct curve O defined by v;
            Check the deviation of curve O from the true offset;
        End;
        Return v;
    End;
```

The value of the offset in the smooth region, d_0, is decided by the tool radius r_t and required machining precision. In this paper, we set $d_0 = r_t$. In the non-smooth region, if the offset value d_1 is set as r_t there is a possibility that the offset curve O at lower level will intersect with the original curve C. In order to avoid this intersection, the value of d_1 should be set higher than r_t.

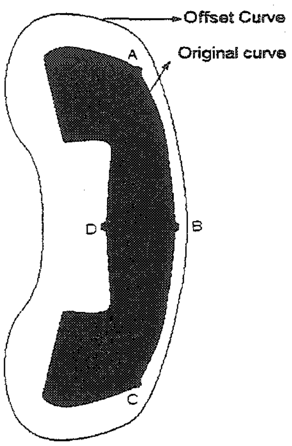


Figure 3.5. Generation of an offset contour for a telephone shaped object.

Different offset distances to smooth regions and non-smooth regions are applied. Smooth region (curve ABC) is offset by distance d_0. Non-smooth region (curve CDA) is offset by distance d_1 (d_1>d_0).

Figure 3.5 shows the generation of an offset contour using the corner point detection and wavelet decomposition for a telephone shaped object. Although different regions of the curve are offset by different amounts, the continuity of the entire offset curve is still maintained. This is because we offset the control polygon of the original curve, and then, generate the offset curve from the new control points. The above algorithms *Tool path curve* and *Offset* are applied repeatedly to the newly generated offset curve to generate subsequent offset curves. These offset curves are the basic tool paths.

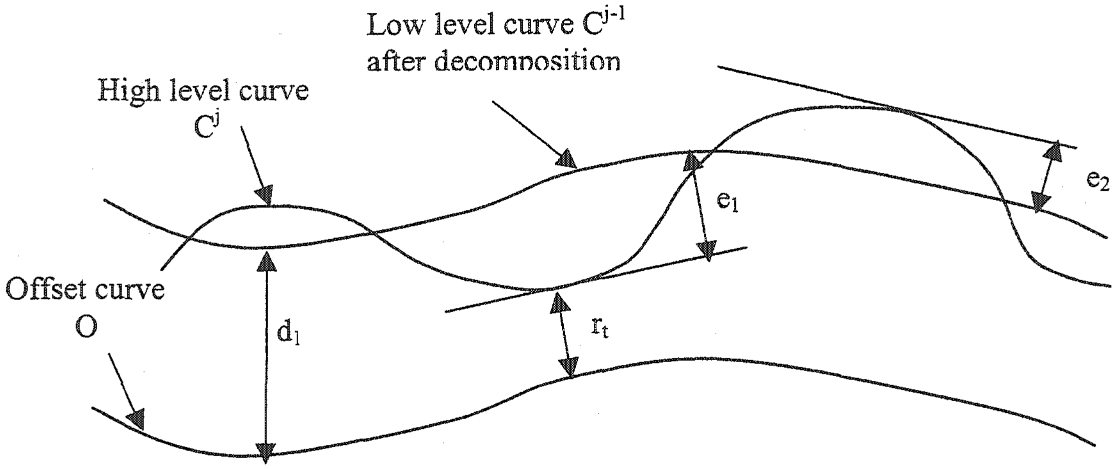


Figure 3.6. Offset distance d$_1$ in non-smooth region.

The value for d_1 may be computed from the stored wavelet coefficients. Figure 3.6 shows an offset curve O to the high-level curve C^j (at level *j*) after wavelet transformation in a non-smooth region. C^{j-1} is the lower level curve at level *j*-1. The wavelet offset curve O can be obtained from C^{j-1} by offsetting a distance $d_1 = r_t + e_1$. As shown in this Figure 3.6, e_1 and e_2 are the maximum deviation (L^∞ error) of C^{j-1} from C^j. The values of e_1 and e_2 may be estimated, by converting the B-spline curve to a set of Bezier curves and using the convex hull properties of the Bezier curve (Bartels *et al.*, 1987). If E^j is defined as a column vector

$$E^j = M^j Q^j D^{j-1} \tag{3.1}$$

Where, M^j is the B-spline to Bezier conversion matrix, Q^j is the synthesis filter as in Equation 2.5 and D^{j-1} is the wavelet coefficients. The vector E^j provides a measure of the distance that the Bezier control points migrate when decomposing the more detailed curve at level *j* to the approximate curve at level *j*-1. Since Bezier curves are contained within the convex hull of their control points, the magnitudes of the entries of E^j provide conservative bounds on approximations to the curve due to truncating wavelet coefficients (Finkelstein and Salesin, 1994).

3.5 Adaptive Tool Path Curves

In non-smooth regions offset distance d_1 is chosen to be larger than the tool radius r_t, to avoid the intersection of the lower level offset curve with the higher-level (detailed) parent curve. Since d_1, the distance between the tool center and the parent curve, is greater than the tool radius, some areas will remain uncut. Adaptive path curves need to be constructed in these non-smooth regions.

The adaptive tool paths are constructed by interpolating between two adjacent basic tool path curves in the non-smooth region. Each non-smooth region N_i^j (scale *j*) on the object contour is progressively decomposed to lower scale curves $N_i^m, m = j-1,\ldots,0$. Interpolation between two adjacent scale curves can be performed by introducing new curves that are at intermediate fractional scales. A curve at the fractional level $m+t$ $(t \in [0,1])$ to interpolate the curve at level m and level m+1 may be obtained as follows.

$$c^{m+t} = P^{m+1}c^{m} + tQ^{m+1}d^{m} \tag{3.2}$$

Where, c^m is the set of control points on a N_i^m, matrices P^{m+1} and Q^{m+1} are the wavelet decomposition matrices as described in Equation 2.16 and c^{m+t} is the newly computed set of control points.

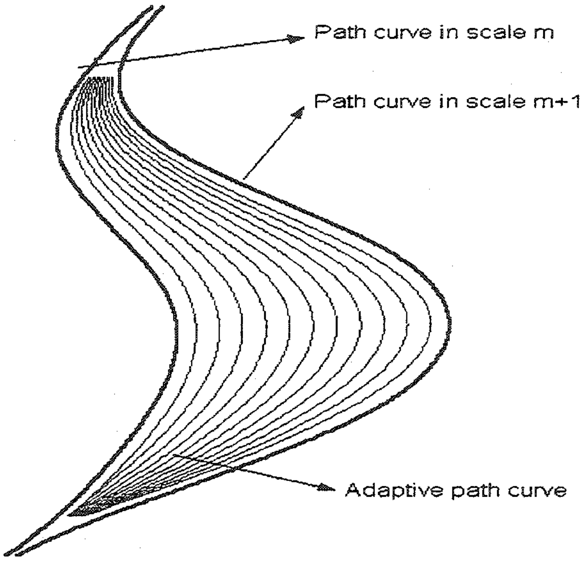


Figure 3.7. Adaptive tool path curves.

A set of interpolating curves between the two non-smooth regions in N_i^m and N_i^{m+1} are obtained when t increases from 0 to 1. These curves are used as adaptive tool path curves. This scheme provides valid coverage of the machining area. The maximum deviations between N_i^m and N_i^{m+1} is given by $e_1 + e_2 + r_t$ as shown in Figure 3.6. Therefore, the number of adaptive curves n that is needed between two adjacent scale curves can be calculated as:

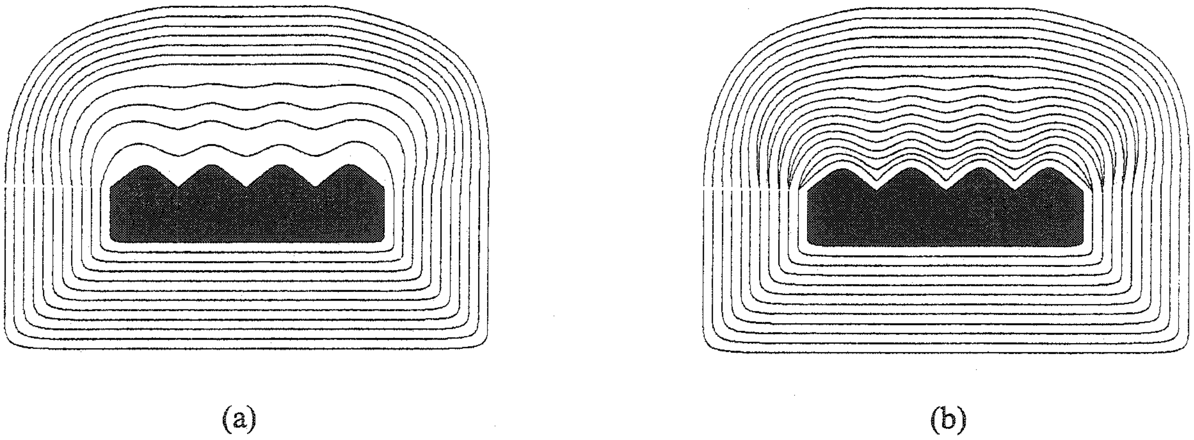
$$n = (e_1 + e_2)/r_t \tag{3.3}$$

Figure 3.7 shows an illustration of adaptive curves based on this technique.

3.6 Tool Path Simulation

The algorithm for tool path generation is implemented using Matlab on a UNIX workstation. Examples are presented to illustrate the tool path generation based on B-spline wavelets. Figure 3.8(a) shows the basic tool path curves for a mountain shaped object. The basic tool path curves resemble the object contour close to the object boundary. The tool path curves become coarser as the distance from the object boundary increases and unnecessary detail features are avoided to improve machining efficiency. In Figure 3.8(b) the adaptive curves are added. Figures 3.9(a) and 3.9(b) show the basic and adaptive tool path curves, respectively, for a telephone shaped object.

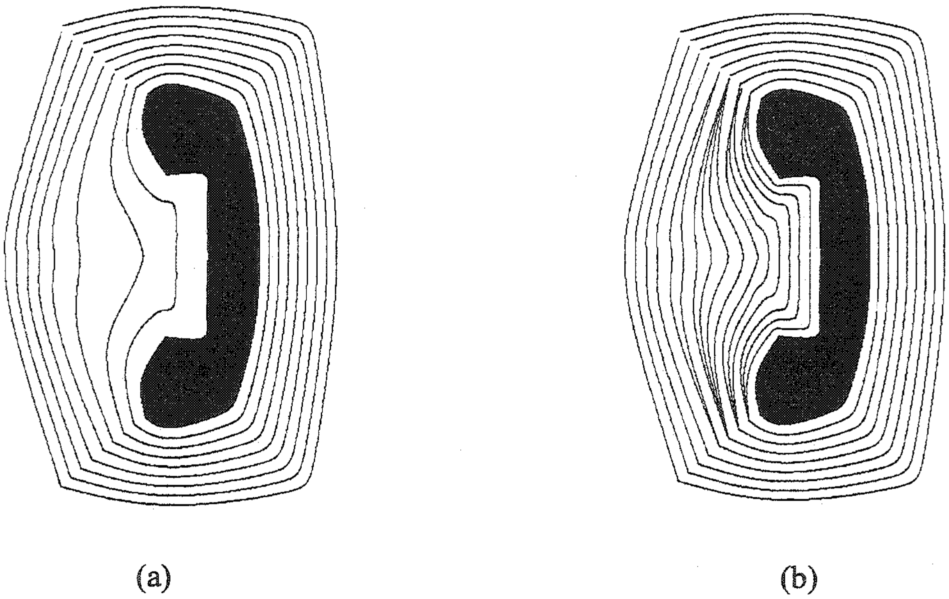
An airplane shaped object and a rabbit shaped object are also shown to illustrate the wavelet based tool path. Figures 3.10 and 3.11 show the tool path. In (a) of Figures 3.10 and 3.11, basic tool path curves are shown. Adaptive tool path curves are added in (b). If necessary, the outer boundaries can be made linear.



(a)

(b)

Figure 3.8. Wavelet based tool path for mountain shaped object.



(a)

(b)

Figure 3.9. Wavelet based tool path for telephone shaped object.

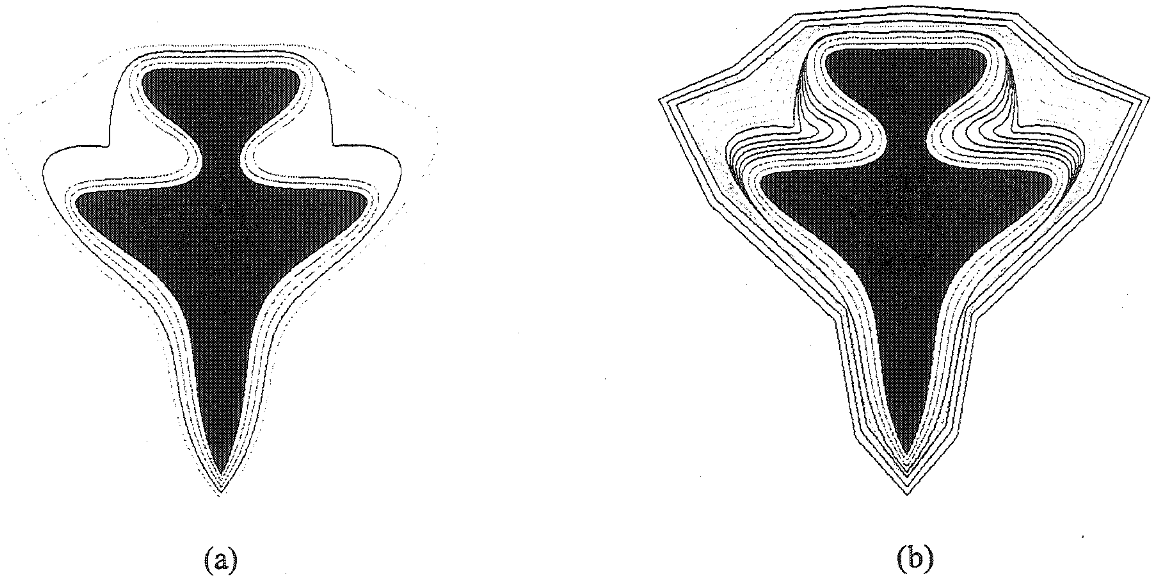


Figure 3.10. Tool path for an airplane shaped object.

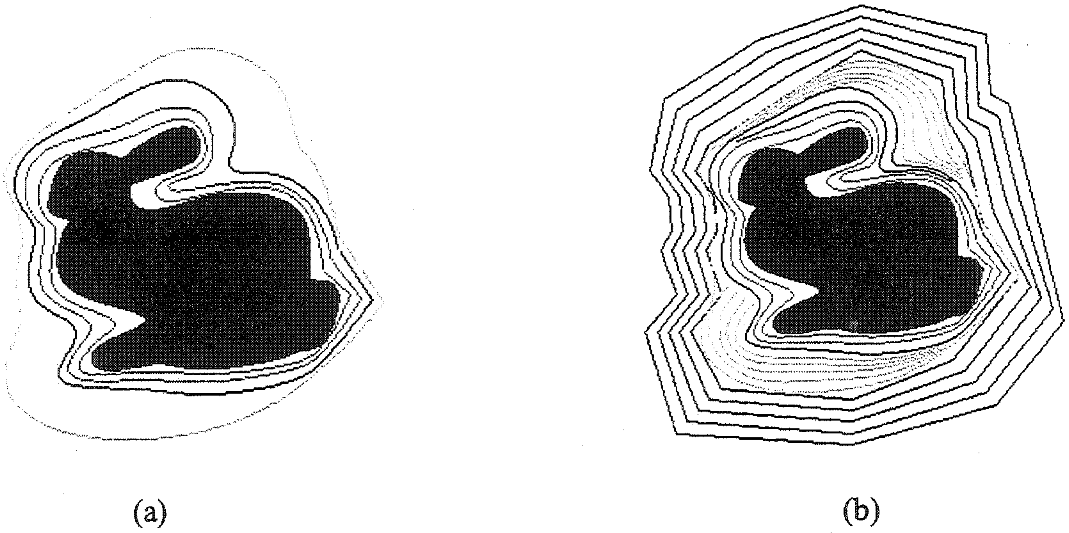


Figure 3.11. Tool path for a rabbit shaped object.

The rabbit and the telephone objects were machined by end milling on a 3-axis Fadal CNC machine using the wavelet based tool path as shown in Figure 3.12. The machined samples confirm that the wavelet-based approach provides an efficient and valid coverage. A ball end mill (0.25in diameter) as opposed to a flat end mill was used to show the cut sequence.

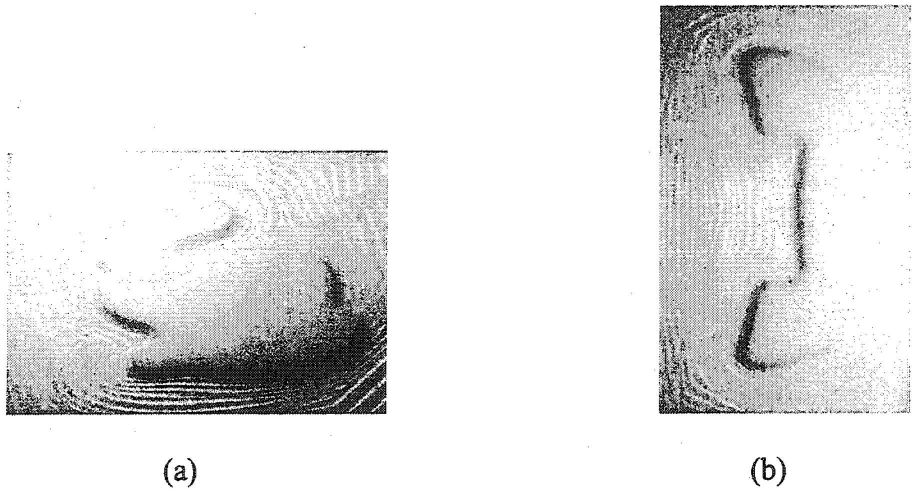


Figure 3.12. Machined samples using wavelet-based tool paths.

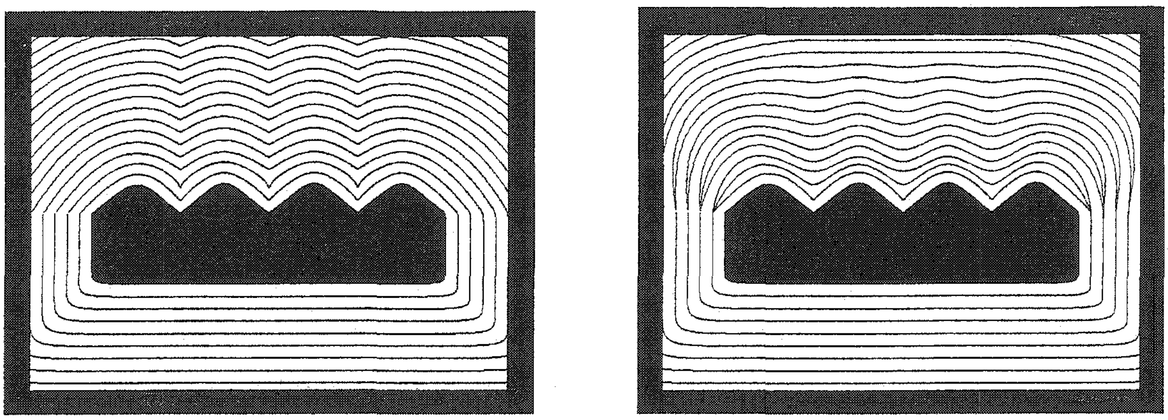
3.7 Discussion

Wavelet theory is a promising technique for tool path generation for NC machining. The cutting efficiency can be increased using coarse and fine curves. In finish cutting, high level or detail curves are used as tool paths for high accuracy. In rough cutting, however, coarse level curves are used. Thus, machining time reduction can be obtained because of the simpler curve geometry at the coarse level.

The traditional contour parallel offset technique and the wavelet-based technique are compared for the mountain shaped object and the telephone shaped object. In the comparison, the machining areas are assumed equal and the machining time is compared using the ProCAM TekSoft-2D machining simulation software.

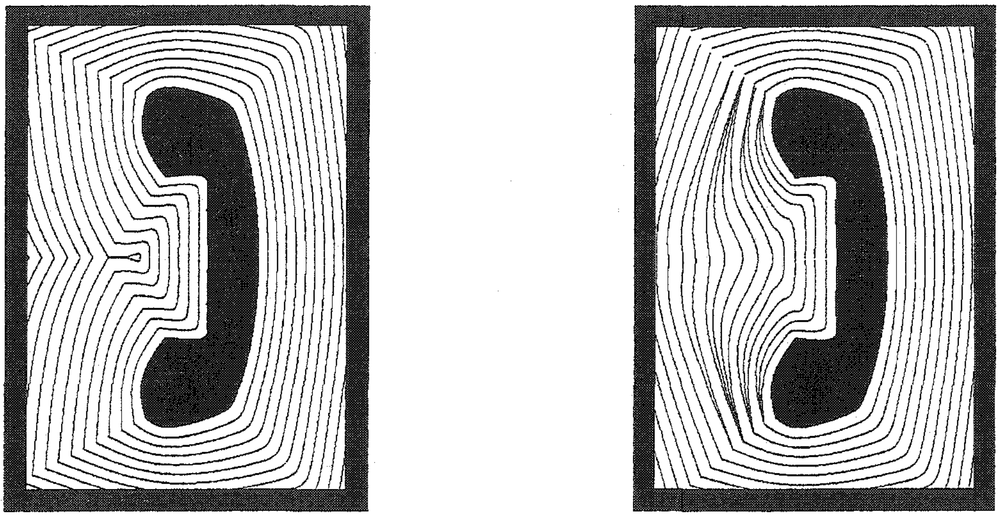
In Figure 3.13(a), the tool path generated by contour parallel method is shown for the mountain shaped object. It can be observed that the same level of detail is maintained at any distance from the object boundary. In Figure 3.13(b) the wavelet based tool path is shown. Here, the tool path curves get coarser farther from the object boundary. This results in improved machining efficiency in two ways. First, the tool path length is reduced compared to the contour parallel offset method. Second, the tool path curves are less complex and the number of discontinuities is reduced. As shown in Table 3.1, the tool path length is reduced from 752.37in in the contour parallel method to 510.19in in the wavelet-based method. This results in a reduction in tool path length of 32%. The machining time in the wavelet-based method is reduced to 51.75 min as compared to 65.03 min in the contour parallel method. This results in a 20% reduction in machining time.

In Figure 3.14, the contour parallel (Figure 3.14 (a)) and wavelet-based tool path (Figure 3.14 (b)) are compared for the telephone shaped object. As shown in Table 3.1, in this case the tool path length is reduced from 914.38in to 835.47in. This is a decrease of 9%. The machining time is reduced from 77.52 min to 71.87 min corresponding to a decrease of 7%.



(a) (b)

Figure 3.13. Machining simulation comparison for mountain shaped object.



(a) (b)

Figure 3.14. Machining simulation comparison for telephone shaped object.

It can be observed that the saving in machining time depends on the type of object considered. The machining time is reduced more in the case of the mountain shaped object compared to the telephone shaped object because the tool path length is reduced by a larger percentage. The percentage reduction in tool path is related to the geometry of the object. In general, the wavelet-based method offers more savings in machining time when the number of non-smooth segments in the object contour is higher.

The adaptive curves can however cause some over-machining. While this problem may be solved using sub-isocurves (Elber and Cohen, 1994), the adaptive curves appear only in the finish cutting stage, and the machining time increase is less noticeable. Tool stops are introduced at the end points of the adaptive curves and can contribute to a small increase in machining time.

Table 3.1. Comparison of wavelet based method and contour parallel method

	Contour parallel method		Wavelet based method	
	Tool path length (in)	Machining time (min)	Tool path length (in)	Machining time (min)
Mountain shaped object	752.37	65.03	510.19	51.75
Telephone shaped object	914.38	77.52	835.47	71.87

(Tool radius is 0.25in, spindle speed is 3500 RPM, XY feedrate is 30 ipm, number of steps in Z-axis is 4. Work-piece size for mountain shaped object is 4in x 6in, and for the telephone shaped object is 5in x 7in).

Smoother tool paths improve machining efficiency by avoiding uncut areas as well. Figure 3.15(a) and Figure 3.15(b) show the contour parallel tool path and wavelet based tool paths, respectively for the mountain shaped object. As noted earlier, the progressively lower detail offset curves are increasingly smoother as we move away from the object boundary. As a result for the same sized tool, the uncut area is either eliminated or reduced for wavelet based machining as compared to contour parallel machining (Figure 3.15).

Smoother curves obtained in wavelet-based machining can also be beneficial for tool selection. Larger sized cutting tools can be used to machine the coarse curves, away from the object boundary, and corresponds to the rough cutting stage. This may not be always possible with current offset techniques, particularly if sharp corners in the object boundary are propagated in each offset curve.

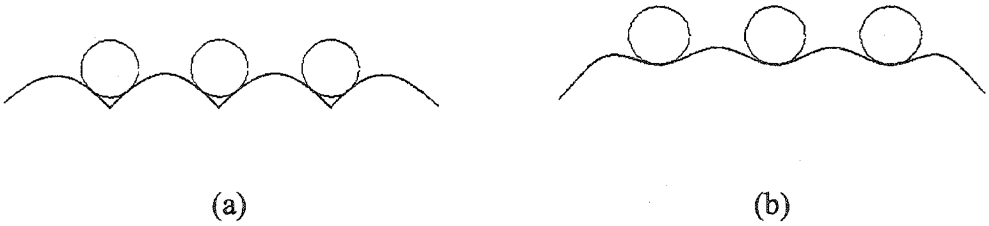


Figure 3.15. Uncut regions that exist in contour parallel method (a) may be reduced in wavelet based method (b).

3.8 Summary

Wavelet based tool path planning was applied to machining 2D contours. An integrated rough and finish cut strategy was developed. The wavelet method results in exact offsets close to the object boundary as in the contour parallel method, but approximated or coarse curves are used away from the object boundary. This results in simpler curves in rough machining. Simulations indicate that the wavelet based tool path lengths are shorter and also simpler in terms of the geometry. Machining time saving of up to 20% was obtained. The smoother curves in wavelet tool paths also reduce the uncut areas. Larger tool sizes may also be selected in rough machining because the tool path is free of any sharp corners.

Wavelet based tool path planning still uses curves as tool paths in rough machining. Interpolation methods are then applied to generate the straight lines or circular arcs for practical machining. In next chapter, we introduce an improved multiresolution tool path generation method based on the convex hull techniques. Compared to the method presented in this Chapter, the improved method improves the rough machining efficiency. Thus, the whole rough-to-fine tool path lengths and machining time are reduced.

CHAPTER 4 IMPROVED MULTIRESOLUTION TOOL PATH PLANNING BASED ON LOOSE CONVEX HULL

In Chapter 3, we introduced a wavelet based multiresolution technique for generating an integrated tool path for rough and fine machining. High-resolution offset curves are used close to the contour while low-resolution offset curves are used farther away from the contour. But, the machining efficiency is not high enough because large number of curved geometries are still used as tool paths, which are still relatively complex for NC machining and will result in large and data-dense G-code program files. In addition, the wavelet curve construction is not easy because knot inserting and deleting are involved. Since most CNC interpolators today accommodate only straight line and circular arc elements, intuitively, the cutting efficiency in the method could be improved by applying straight lines and circular arcs as tool paths, especially in rough cutting.

Chuang and Lin (1997) developed a method, based on the convex hull property of Bezier curves, in which the portion of the contour defined by a Bezier curve is replaced by parts of its convex hull on the pocket side. Therefore, the new pocket boundary is redefined by only segments of straight lines. Furthermore, recursive subdivisions on Bezier curves are used to improve the cutting accuracy with an allowance criterion based on sizes of convex hulls. However, the line segments approximating curve can only achieve G^0 continuity. After the approximate line segments are offset for tool path generation, the machining paths cause discontinuity and sharp corners at joints of line segments. Chuang and Kao (1999) improved above method by applying biarc fitting and single arc fitting method to construct a smooth, G^1 continuous curve. Thus, sharp corners and abrupt direction change on tool paths can be greatly reduced.

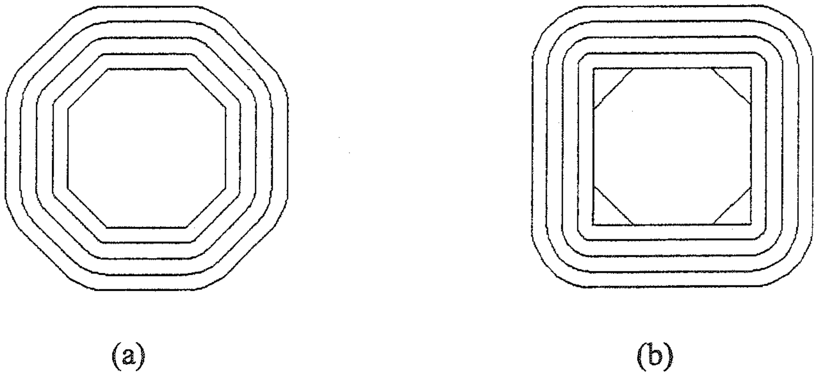
In Chuang and his collegues' method, convex hulls are used to avoid overcut. In fact, convex hulls could be used to construct rough cutting paths. The convex hull is an approximate representation of original Bezier curve or B-spline curve. If we construct tool paths in rough cutting by offsetting the convex hull of the original curve. The geometric shape of the original contour curve would be kept to some degree, and most important, the cutting efficiency could be improved because only straight lines are used. No other complex curved geometries, and no further interpolation computation is needed.

In this Chapter, we propose a tool path generation method based on the multiresolution analysis and convex hull techniques. This is an expended work from our previous work in Chapter 3.

The rest of the chapter is organized as follows. In section 4.1, the loose convex hull and its offset technique are derived. In section 4.2, wavelet curves clipping is introduced. Tool path simulations are presented in section 4.3. In section 4.4 the wavelet and convex hull based tool path is compared with the contour parallel method. Conclusions are presented in section 4.5.

4.1 Loose Convex Hull

The B-spline curve has the convex hull property, which is the B-spline curve lies in the convex hull of the control points. The convex hull of the control points is, to some degree, an approximate representation of the original curve. The convex hull and its offsets conceptually can be used as tool paths during rough cutting. The number of straight line segments (legs) in the convex hull polygon will affect the efficiency of rough cutting. As shown in Figure 4.1, The tool paths in Figure 4.1(a) and (b) cover the same areas, and the tool paths in (b) have less corner points compared to (a). Generally, the tool paths in (b) results in less machining time compared to those in (a), because of less dwell time in practical machining. Thus, in order to improve the machining efficiency, The number of legs in original convex hull polygon probably need to be reduced. The new convex hull polygon after reducing the number of legs is called as the *loose convex hull.* The loose convex hull has less number of edges, and is required to be approximate to the original convex hull as much as possible. In the following, we give a simple algorithm *LooseConvexHull* to construct a loose convex hull from its original convex hull.



(a)

(b)

Figure 4.1. Machining time and the number of edges in convex hull.

```
Algorithm LooseConvexHull ()
Begin
    Construct the convex hull of B-spline shape curve;
    Set threshold value $L$;
    $i := 1$;
    $k := 0$;
    $u := 0$;
    While $i \le n$ (number of legs) do
        Compute the length $l_i$ of leg $i$;
        While $l_i \le L$ do
            $k := k + 1$; ( $k$ is the number of legs whose length is less than $L$)
            $s_k := l_i$;
        End;
        $i := i + 1$;
    End
    $i := 1$;
    While $i \le k$ do
        If $s_i$ is not adjacent to $s_{i+1}$, Then
            Delete $s_i$;
        Else
            $u := u + 1$; ( $u$ is the number of combined legs)
            Combine $s_i$ and $s_{i+1}$ into $t_u$;
        End If;
        $n := u$;
        $l_i = t_u$;
        LooseConvexHull ();
    End;
End;
```

In above algorithm, we delete short leg s_i if it is not adjacent to any other short legs, as shown in Figure 4.2(a), or we combine short legs s_i and s_{i+1} if they are adjacent to generate a new leg t_u, as shown in Figure 4.2(b) for adjacent two short legs. First, the bi-sector line is used in the intersecting point between two adjacent short legs s_i and s_{i+1}. A straight line perpendicular to the bi-sector line is intersected to the adjacent two edges to generate a new leg t_u. The methods of deleting and combining more than two adjacent short legs are similar. Figure 4.2(c) and (d) shows the situations for three and four adjacent short legs, respectively.

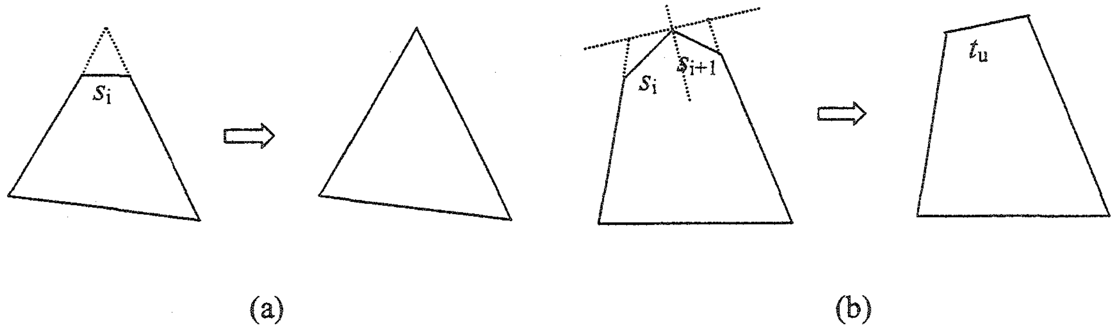


Figure 4.2. Loose convex hull generation.

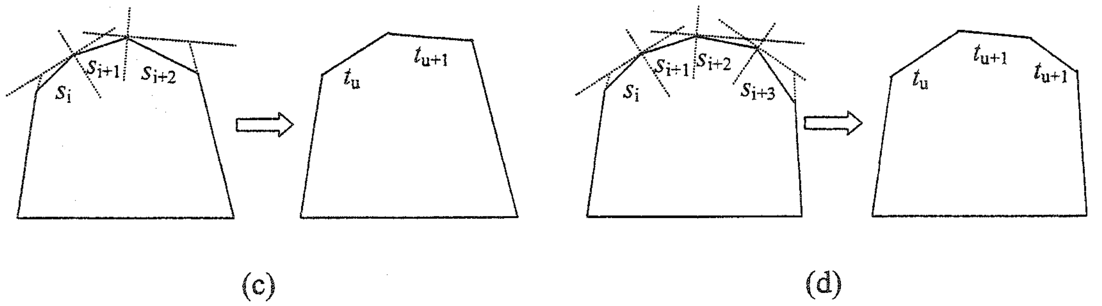


Figure 4.2. Continued

Figure 4.3 at below is the original convex hull (Figure 4.3(a)) and loose convex hull (Figure 4.3(b)) for a phone shaped object. Only deleting operation is involved to construct the loose convex hull of the shape.

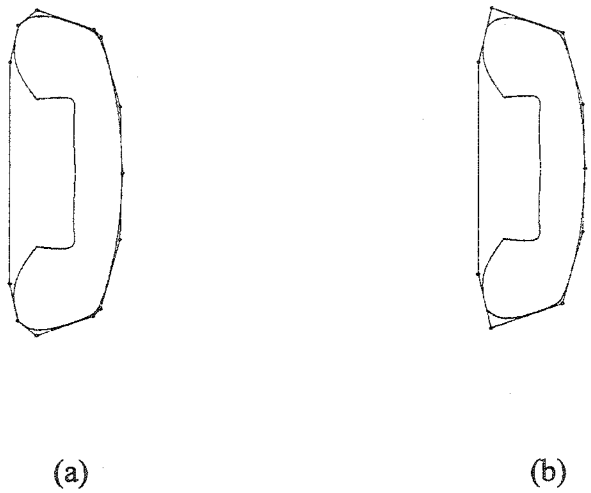


Figure 4.3. Convex hull for the phone shaped object.

The loose convex hull will be used to generate rough cutting path. To do this, first we simply offset each straight line leg of the loose convex hull by an offset distance *r*, which is the tool radius. Then, adjacent offset straight lines are connected by a circular arc, whose radius is *r* and center is the intersection point of the two straight line legs in original loose convex hull. The offset is illustrated in Figure 4.4.

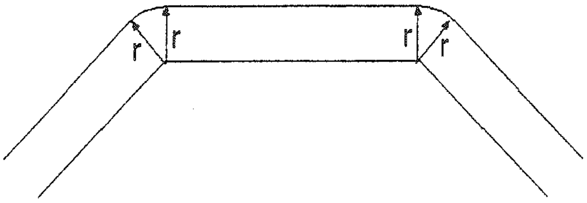


Figure 4.4. Convex hull offset

The loose convex hull is offset successively to form the tool path for rough machining. The loose convex hull offset for the phone shaped object is shown in Figure 4.5(a). Figure 4.5(b) shows the successive offsets as tool paths.

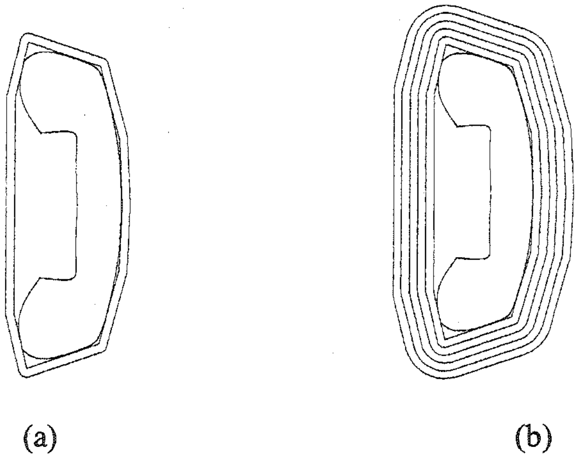

(a)
(b)

Figure 4.5. Loose Convex hull offset for the phone shaped object.

4.2 Wavelet Curve Clipping

In the region between initial object contour and its loose convex hull, Wavelet based multiresolution curves, as described in Chapter 3, can be used as fine tool paths for 2.5D machining,

The generation of tool paths for 2.5D objects based on wavelets broadly involves the following steps as described in Chapter 3:

1. Detect corner points in the initial contour based on curvature analysis;
2. Partition the whole contour into smooth regions (low density) and non-smooth (high density) regions;
3. Wavelet decompose non-smooth regions to adjacent lower level;
4. Offset the contour based on wavelets. Smooth regions and non-smooth regions are offset with different distances;
5. Repeat steps 3 and 4 to current offset curve to generate the machining path sequence progressively outwards;
6. Add adaptive path curves between adjacent offset curves in non-smooth regions.

After wavelet based tool paths are generated near object boundary, the loose convex hull, actually acting as a convex polygonal clipping window, clips away all wavelet curves segments lying outside the loose convex hull polygon. The clipping process is actually B-spline curve clipping against convex polygonal window.

The curve clipping algorithm relies on the convex hull property and the subdivision property of the curve (the de Casteljau algorithm). First, we transform the B-spline curve to the Bezier curve, because the Bezier curve has local convex hull property, and the convex hull is tighter than that of B-spline, which are better for subsequent computation. Figure 4.6 shows the convex hull of a single Bezier curve, and the convex hulls after subdividing into two and four pieces.

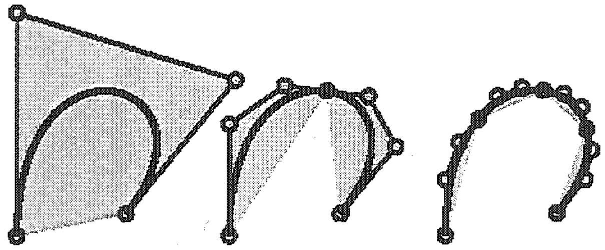


Figure 4.6. Convex hull and subdivision

The clipping algorithm proceeds by comparing the convex polygonal window and the convex hull of each 4-control-point (for cubic case) Bezier curve segment. If they do not overlap, the curve lies outside the window, and should be clipped away. If they do overlap, the curve is subdivided and the two halves of the curve are checked for overlap against the convex polygonal window. As this procedure continues, each curve segment whose convex hull lying inside the window is kept as visible. Once a curve has been subdivided enough that it can be approximated by a line segment to within a tolerance ε, the intersection of the curve and the clipping window is also found by tracing the overlapping.

The clipping algorithm is briefly described in Algorithm *Clip* as follows. The original B-spline curve is first transformed to 4-control-point Bezier curve segments c_1, c_2..., and c_n. Then, the *Clip* is applied to each of the Bezier curve segments.

```
Algorithm Clip ($c$) ($c$ is a 4-control-point Bezier curve segment)
Begin
    While $i \le n$ do
        Compute the convex hull $C$ of the Bezier curve $c$;
        Overlap determination between $C$ and the clipping window;
        If not overlap then
            Clipped away the curve;
        Else If C is contained in the window totally
            Keep the curve visible;
        Else
            While the tolerance between the approximated line and the curve $> \varepsilon$ do
                Subdivide the $c$ into $c_1$ and $c_2$;
                Clip ($c_1$);
                Clip ($c_2$);
            End;
            Compute the intersection point between the curve and the window;
        End If;
    End;
End;
```

To do the *overlap determination*, we adapted a simple linear algorithm developed by Toussaint (1985). Let *P* and *Q* be two convex polygons, whose intersection is a convex polygon. The Toussaint's algorithm for finding this convex intersection polygon can be described by three steps:

1. Construct the convex hull of the union of *P* and *Q* using the *rotating calipers* method;
2. Identify the segments of *P* and *Q* which intersect. By a special-purpose triangulation procedure for sail polygons (a polygon composed of two concave chains sharing a common vertex at one extremity, and connected by a segment (the *mast*) at the other end), The step can be implemented in linear time. $O(k+l)$, where k and l are the number of vertices of *P* and *Q* which are inside the sail polygon.
3. Merge together the polygonal chains between the intersection points found in step 2.

In our *overlap determination*, step 3 is not needed. In fact, the *overlap determination* stops as soon as one intersecting segment on any one of the two polygons is identified.

All of this assumes that the Bezier convex hull and the clipping window do intersect. However, there are two situations in which no intersection exist:

1. The Bezier convex hull and the clipping window simply do not intersect each other and are separable (described as “not overlap” in algorithm *Clip*);
2. The Bezier convex hull is contained in the clipping window.

For case 1, this is detected if the Toussaint’s algorithm, during triangulation in step 2, makes a complete loop around one of the polygons. Case 2 is even easier to detect , in such a case no new convex hull will be found in step 1.

In the *Clipping* algorithm, x-y bounding boxes can be used to substitute the convex hull to greatly simply the convex hull computation and overlap determination. The x-y bounding box can be obtained from the control net of the curve. The four minimum and maximum x- and y-coordinates in the control net are used to construct the bounding box.

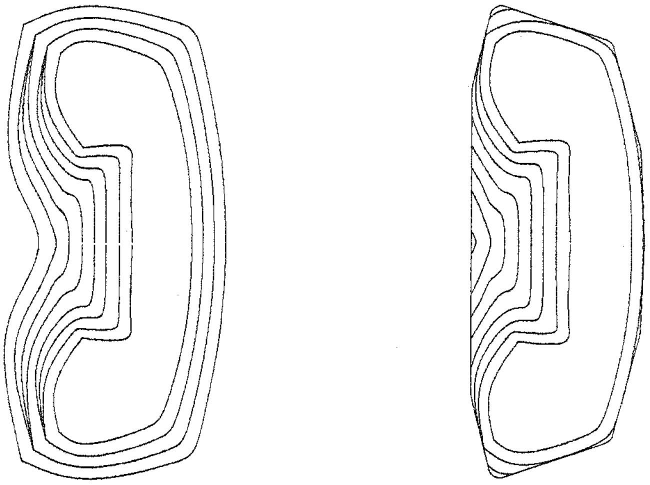


Figure 4.7. Curve clipping against convex polygonal window.

The algorithm *Clip* is applied to all wavelet curves to obtain the tool paths between the object contour and its loose convex hull. Figure 4.7 shows the clipping for telephone shaped object. Figure 4.7(a) shows the wavelet curves before clipping, (b) shows the curves after clipping.

In fact, if we can determine how many wavelet curves are needed to do the clipping, the computation to construct the wavelet curve can be saved to a great degree. To determine the maximum number, denoted by n_c, of wavelet curves needed, we can find the circumscribed circle of the loose convex hull. Note that we refer the circumscribed circle here to the circle which includes the loose convex hull inside with the minimum radius, instead of the circumscribed circle in exact mathematic meaning because there is no such circle for any polygon.

In order to guarantee that all regions between the object contour and the loose convex hull are covered by wavelet curves, the n_c can be computed by:

$$n_c = \lceil r_c / r - 1 \rceil \tag{4.1}$$

Where r_c is the radius of the circumscribed circle, r is the offset distance, $\lceil\ \rceil$ represents taking the integer larger than the result inside. The equation gives a conservative computation for any object shape, and works well for our experiments introduced later.

4.3 Tool Path Simulation

The algorithm for tool path generation is implemented using Matlab on a UNIX workstation. Examples are presented to illustrate the tool path generation based on loose convex hull offsets and B-spline wavelets. Figure 4.8(a) shows the convex hull offsets as rough tool paths in a given rectangular region for the phone shaped object. Straight lines and circular arcs are used, and unnecessary detail features are avoided to improve machining efficiency. In Figure 4.8(b) the wavelet curves are added. Figure 4.9 shows the rough tool paths using convex hull offsets (Figure 4.9(a)) and fine tool paths using wavelet curves (Figure 4.9(b)) for an "up" in Chinese shaped object.

The flower shaped object is also shown to illustrate the convex hull and wavelet based tool path. Figures 4.10 shows the tool path. In (a), convex hull offsets are generated. In (b), wavelet curves are added.

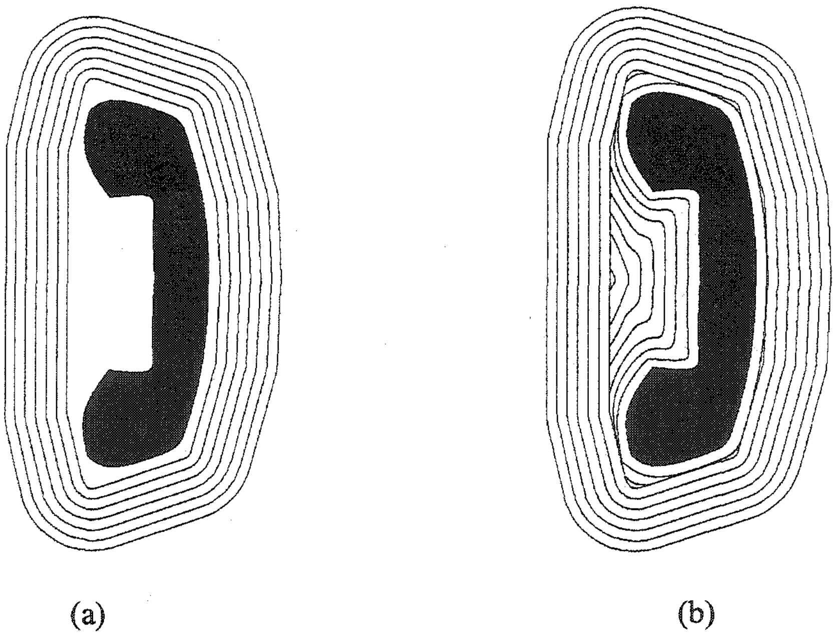


Figure 4.8. Tool paths for telephone shaped object.

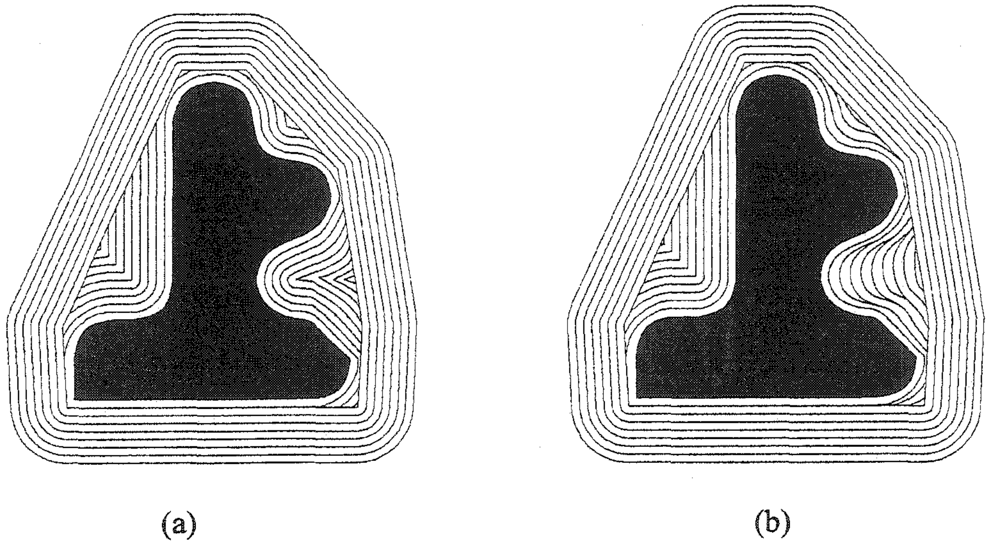


Figure 4.9. Tool paths for “up” in Chinese.

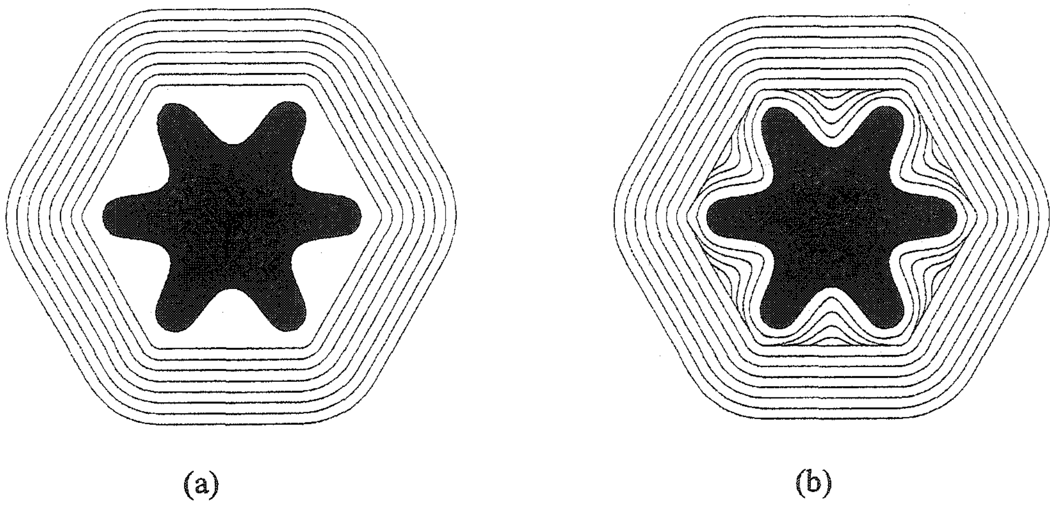


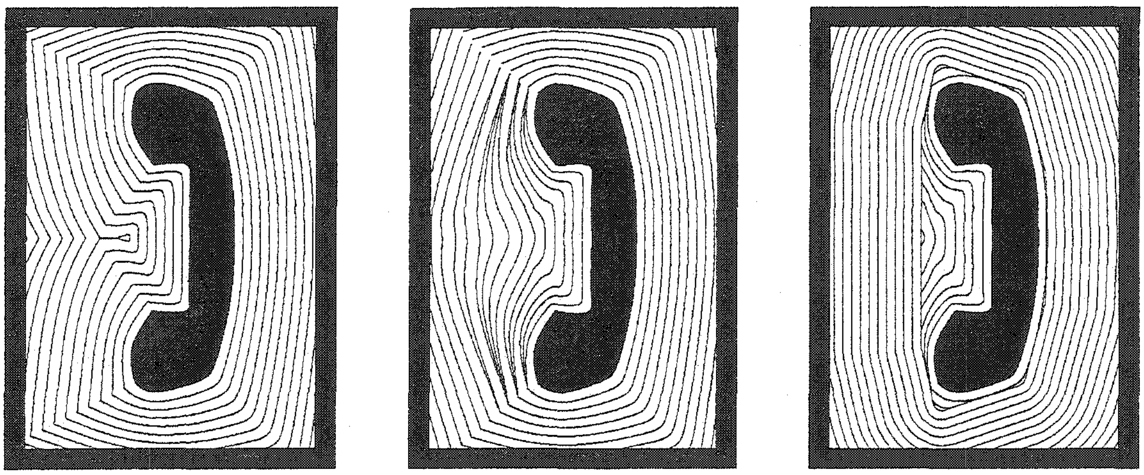
Figure 4.10. Tool paths for flower shaped object.

Instead of offsetting the loose convex hull, morphing techniques can be used to construct the tool path curves between the initial loose convex hull and the outer rectangular border by smoothly blending between the loose convex hull and the rectangular border.

4.4 Discussion

Wavelet theory combined with the convex hull approach is a promising technique for tool path generation for NC machining. The cutting efficiency can be increased using coarse and fine tool path curves. In finish cutting, detail curves at different levels are used as tool paths for high accuracy. In rough cutting, however, straight lines and circular arcs are used. Thus, machining time reduction can be obtained because of the simpler tool path geometry at the coarse level.

The wavelet and convex hull based technique is compared with the traditional contour parallel offset technique and wavelet based technique, respectively, for the phone shaped object. In the comparison, the machining areas are assumed equal and the machining time is compared using the ProCAM TekSoft-2D machining simulation software. The data for the traditional contour parallel offset technique and the wavelet-based technique are taken from Table 3.1 in Chapter 3.



(a) (b) (c)

Figure 4.11. Machining simulation comparison between contour parallel (a), wavelet based method (b), and wavelet and convex hull based method (c).

In Figure 4.11(a), the tool path generated by contour parallel method is shown. It can be observed that the same level of detail is maintained at any distance from the object boundary. In Figure 4.11(b) the wavelet based tool path is shown. Figure 4.11(c) shows the tool path generated by the wavelet and convex hull based method. Here, the tool path curves get coarser farther from the object contour as in wavelet based method. But, outside the loose convex hull, the convex hull offsets are used as tool paths. This results in improved machining efficiency in three ways. First, the tool path length is reduced compared to other methods. Second, the fine tool path curves are less complex and the number of discontinuities in fine tool path is reduced compared to the contour parallel method. Third, the straight lines and circular arcs are very convenient and fast for high speed machining in the CNC machine.

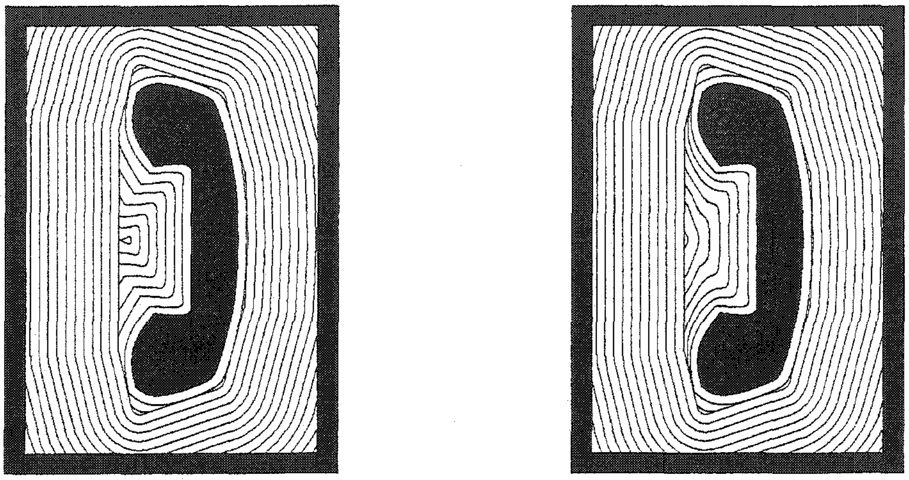
As shown in Table 4.1, the tool path length is reduced from 914.38in in the contour parallel method to 835.47in in the wavelet-based method. The length is further reduced to 792.53 in the wavelet and convex hull based method. This results in a reduction in tool path length of 13% compared to the contour parallel method and 5% compared to the wavelet based method, respectively. The machining time in the wavelet and convex hull based method is reduced to 60.74 min. This results in a 21% reduction in machining time as compared to 77.52 min in the contour parallel method, and a 15% reduction as compared to 71.87min in the wavelet based method.

Table 4.1. Comparison for telephone shaped object

Contour parallel method		Wavelet based method		Wavelet and convex hull based method	
Tool path length (in.)	Machining time (min)	Tool path length (in.)	Machining time (min)	Tool path length (in.)	Machining time (min)
914.38	77.52	835.47	71.87	792.53	60.74

In Figure 4.12, the wavelet and convex hull based tool path is further compared with the contour parallel and convex hull based method for the phone shaped object. As shown in

Table 4.2, in this case, for the phone shaped object, the tool path length 792.53in and the machining time 60.74min in the wavelet and convex hull based method is a reduction of 1.9% and 3%, respectively, compared to 807.95in and 62.37min in the contour parallel and convex hull based method.



(a)

(b)

Figure 4.12. Machining simulation comparison for telephone shaped object between contour parallel and convex hull based method (a), and wavelet and convex hull based method (b).

Figure 4.13 shows the comparison for the “up” in Chinese shaped object. The tool path length is reduced from 814.33in in the contour parallel and convex hull based method to 797.14in in the wavelet and convex hull based method. This is a decrease of 2.1%. The machining time is reduced from 67.52 min to 64.87 min corresponding to a decrease of 4%.

In Figure 4.14, we also show the comparison for the flower shaped object. The tool path length is reduced from 908.22in in the contour parallel and convex hull based method to 878.05in in the wavelet and convex hull based method. This is a decrease of 3%. The machining time is reduced from 73.59 min to 66.32 min corresponding to a decrease of 9%.

Table 4.2. Tool path length and machining time comparison

	Contour parallel and convex hull based method		Wavelet and convex hull based method	
	Tool path length (in.)	Machining time (min)	Tool path length (in.)	Machining time (min)
Telephone shaped object	807.95	62.37	792.53	60.74
"Up" in Chinese	814.33	67.52	797.14	64.87
Flower shaped object	908.22	73.59	878.05	66.32

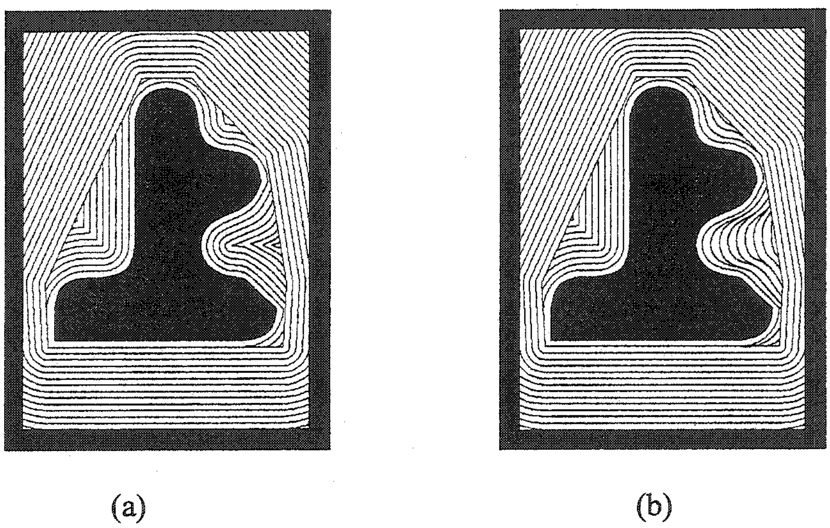


Figure 4.13. Machining simulation comparison for "up" in Chinese between contour parallel and convex hull based method (a), and wavelet and convex hull based method (b).

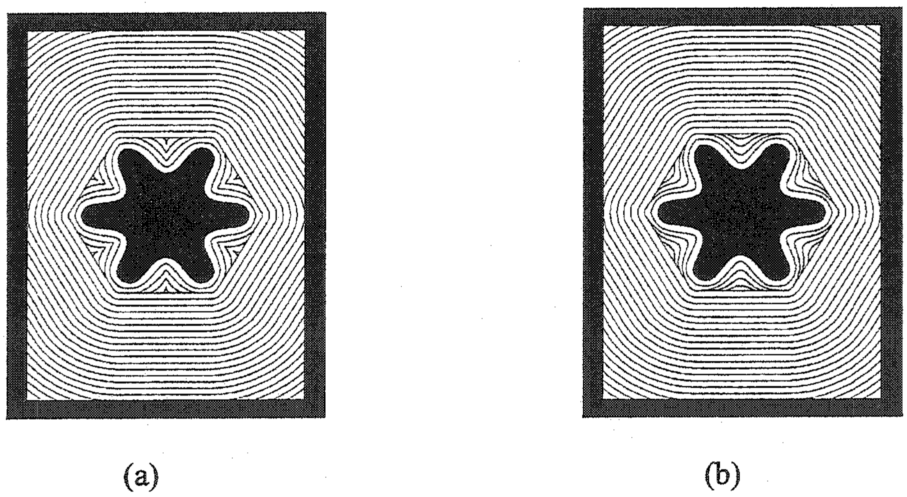


Figure 4.14. Machining simulation comparison for flower shaped object between contour parallel and convex hull based method (a), and wavelet and convex hull based method (b).

It can be observed that the saving in machining time depends on the type of object considered. The machining time is reduced more in the case of the flower shaped object compared to the telephone shaped object because smooth tool path curves are used in a larger percentage. The percentage reduction in tool path is related to the geometry of the object. In general, the wavelet and convex hull based method offers more savings in machining time when the number of concave regions in the object contour is higher.

Simple straight lines, circular arcs and smoother curves obtained in wavelet and convex hull based machining can also be beneficial for tool selection. Larger sized cutting tools can be used to machine the coarse curves, away from the object boundary, and corresponds to the rough cutting stage. This may not be always possible with current offset techniques, particularly if sharp corners in the object boundary are propagated in each offset curve.

4.5 Summary

B-spline wavelet based multiresolution analysis techniques and convex hull method are applied to generate tool paths for 2D contour machining. An integrated rough and finish cut strategy was developed. High-resolution offset curves are used close to the object boundary while low-resolution offset curves; straight lines and circular arcs are used farther away from the boundary. This results in simpler tool paths for rough machining. Simulations indicate that the new tool path generation method reduces the tool path length and machining time to a great degree.

In next chapter, we will provide the basic concept and method about applying multiresolution analysis and wavelet theory in five-axis NC machining. Especially, multiresolution offset technique and multiresolution accessibility analysis will be the focus of our research work.

CHAPTER 5 FIVE-AXIS MACHINING

Using the second generation wavelet, a complex surface can be represented at a very high resolution with very small triangular facets (Lounsbery *et al.*, 1997). At lower resolutions fewer triangles with a larger size are used.

Wavelets and multiresolution analysis may be used to hierarchically represent the original surface. Each hierarchical model successively approximates the final object shape. Thus rough-cuts can be made with the low-resolution models and finish cuts can be made with the high-resolution models. Wavelets based multiresolution analysis may also be used to develop new surface offset method for subdivision mesh surface. Multiresolution analysis and wavelet theory also have potential in accessibility analysis for efficient collision detection and avoidance in 5-axis machining.

For our work on wavelet-based 5 axis NC tool path generation we assume that: (i) the final surface of a work piece is represented by a mesh; (ii) The setups and fixtures have already been selected; and (iii) A stock is oriented in some pre-specified way relative to fixtures in a setup. Our approach consists of the following steps:

(1). Surface Meshing

The multiresolution representation of a mesh using subdivision wavelets has a prerequisite that the input mesh is a semi-regular mesh, that is, most vertices of the mesh have fixed valence (same as for a regular grid), and only few isolated vertices have a different valence.

The goal of this step is to evaluate current semi-regular mesh generation methods from CAD data or reverse engineered data.

(2). Wavelet Decomposition

Wavelet based multiresolution analysis will be used to convert the original meshed surface with arbitrary topology to its multiresolution representation (Figure 5.1). Figure 5.1(b) shows the low-resolution representation of the surface in Figure 5.1(a) by implementing a low pass filter A. The detail part, consisting of wavelet coefficients, is obtained by a high pass filter B. The decomposition process, called analysis, further splits Figure 5.1(b) into an even lower resolution version and corresponding wavelet coefficients (Figure 5.1(c)). The filter bank algorithm culminates with the coarsest level representation in Figure 5.1(d), together with wavelet coefficients at each level.

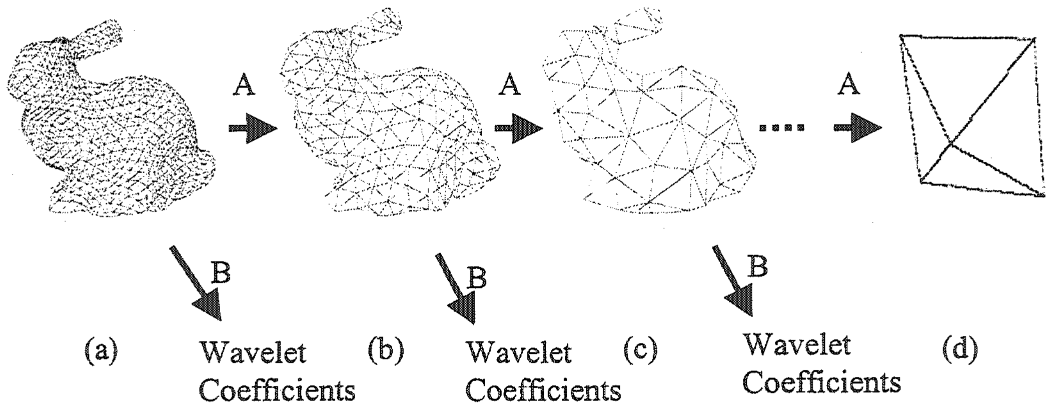


Figure 5.1. Decomposition of a mesh

Among the variety of existing surface subdivision schemes, we choose Loop subdivision scheme in our research. The Loop scheme is a simple approximating face-split scheme for triangular meshes proposed by Charles Loop (1987). The Loop scheme produces surfaces that are C^2-continuous everywhere except at extraordinary vertices, where they are C^1-continuous. The subdivision surface $S(M)$ associated with a control mesh M is defined as the limit of a refinement process applied to *M*:

$$M, \quad M^1 = R(M), \quad M^2 = R(R(M)), \quad \ldots$$

Where the refinement procedure *R* proceeds by splitting each triangular face into four surfaces. The vertices of the refined mesh are then positioned using weighted averages of the vertices in the unrefined mesh. Formally, starting with the initial control mesh $M = M^0$, each subdivision step carries a mesh M^r into a refined mesh M^{r+1}. The vertices V^{r+1} in M^{r+1} are computed as affined combinations of the vertices of V^r in M^r.

Some of the vertices of V^{r+1} naturally correspond to vertices of V^r -these are called vertex points; the remaining vertices in V^{r+1} correspond to edges of the mesh M^r -these are called edge points. Let v^r denote a vertex of V^r having neighbors v_1^r, …, v_n^r as shown in Figure 5.2. Such a vertex is said to have valence n. Let v_i^{r+1} denote the edge point of V^{r+1} corresponding to the edge $v^r v_i^r$, and let v^{r+1} be the vertex point of V^{r+1} associated with v^r. The positions of v^{r+1} and v_i^{r+1} are computed according to the subdivision rules

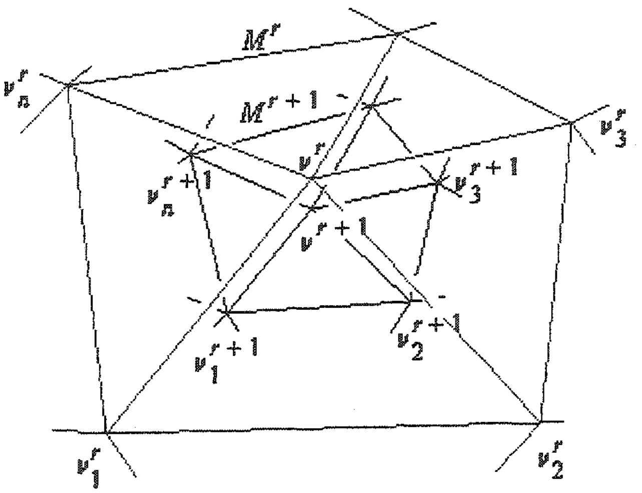


Figure 5.2. The neighborhood around a vertex v^r of valence n.

$$v^{r+1} = \frac{\alpha(n)v^r + v_1^r + \ldots + v_n^r}{\alpha(n) + n}$$

$$v_i^{r+1} = \frac{3v^r + 3v_i^r + v_{i-1}^r + v_{i+1}^r}{8}, \qquad i = 1, \ldots, n \tag{5.1}$$

Where $\alpha(n) = \frac{n(1-a(n))}{a(n)}$ with $a(n) = \frac{5}{8} - \frac{(3+2\cos(\frac{2\pi}{n}))^2}{64}$. Affined combinations such as those in Equation 5.1 can be visualized by diagrams called masks, as shown in Figure 5.3.

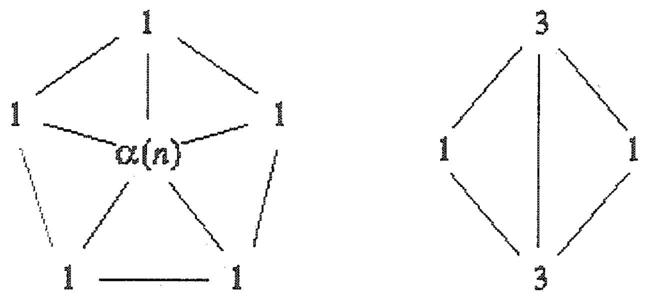


Figure 5.3. Vertex and edge subdivision masks for Loop's subdivision surface scheme.

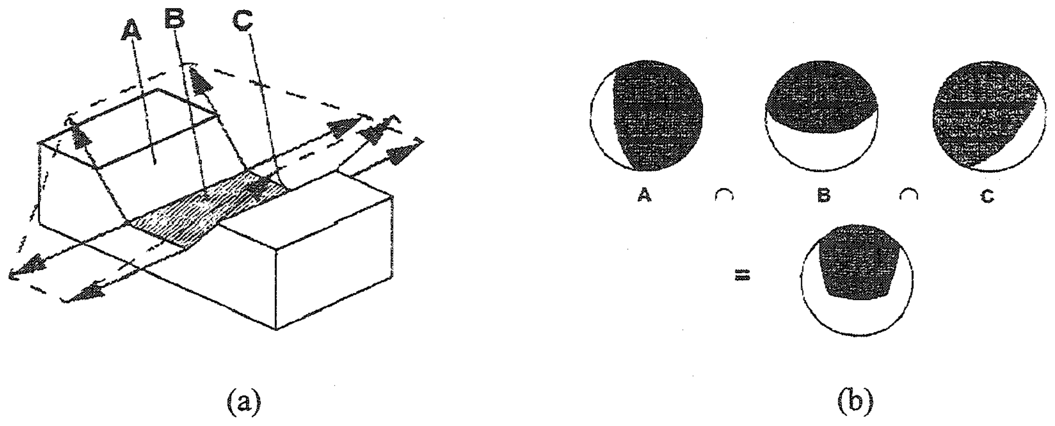
(3). Surface Offset

Obtain intermediate surfaces for machining at different levels of detail. The generation of an intermediate surface is a very important component since the accuracy of CNC tool paths is directly related to the accuracy of the intermediate surface. An issue arises when using these simplified surfaces for machining. They are probably intersecting with the original mesh. To avoid gouging in machining, surface offsetting is necessary. Moreover, to apply the multiresolution surfaces in machining, the surface of a higher resolution model must be covered with that of a lower resolution model in order to prevent overcut.

In this part, an effective and practical offset surface generation method for multiresolution meshed surfaces will be developed. An offset error analysis method based on wavelets will be developed first, in which the multiresolution control method based on the lifting scheme will be used to control offset error to prevent overcut. The value of the wavelet coefficient indicates the difference between a data value that corresponds to the wavelet coefficient and the average of its neighboring data values. Lifting scheme can be used to decide the position of the low-resolution data that are the neighboring points of a higher resolution point corresponding to the wavelet coefficient.

(4). Multiresolution accessibility analysis

Perform accessibility analysis hierarchically. In this stage, we use the concept of visibility to determine from which directions a point in the offset surface is likely to be accessible to a tool located outside the convex envelop of the object. In our research, visibility information is represented using visibility maps (Chen and Woo, 1992). Visibility cones, which are clusters of visibility directions for points on a work-piece, can be mapped on to the unit sphere to create a visibility map, as shown in Figure 5.4. In Figure 5.4(a), visibility cone for the shaded region is shown; (b) shows the visibility information for the concave region, formed by surfaces A, B and C, is given by the intersection of the individual visibility hemispheres of its neighboring faces.



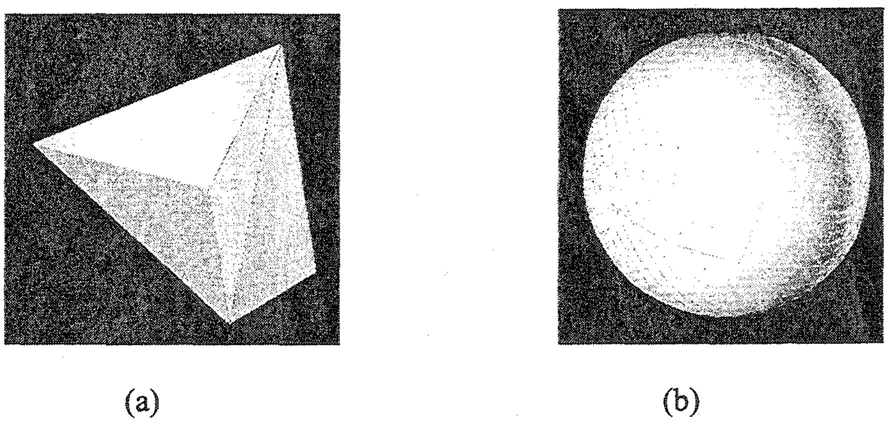
(a)

(b)

Figure 5.4. Visibility cone and visibility maps.

We will develop visibility maps calculation software for triangular meshed surface. A simple example is shown in Figure 5.5. In Figure 5.5(a), the region in shaded color is a concave region formed by four triangular planes; Figure 5.5(b) shows the visibility map for the concave region.

To determine visibility at any resolution, we propose to use visibility information from the lower resolution representations. Wavelet coefficients are used in the procedure. After visibility analysis, the diameter of the tool and the tool holder has to be considered to ensure that the tool can access a required point without any interference


(a)

(b)

Figure 5.5. Visibility map for subdivision meshed surface.

(5). Tool *orientation determination*

For a simple convex mesh, the normal direction of every small triangular plane in the mesh, intuitively, can be used to decide the tool orientation for 5-axis machining. For a complex surface of general topology, the task of tool orientation determination is a local search in the neighborhood of the direction first suggested by visibility analysis. The search criterion is to find the point within each region that results in minimum cusp height. For every sample point in the surface, a set of valid tool orientations will be output after this stage.

(6). Tool Selection

Obtain an optimal set of tool sizes that are needed to machine the object based on the multiresolution representation. Part of tool size information at sample points will be obtained from (4) *multiresolution accessibility analysis* and (5) *tool orientation determination*. The criteria for tool selection are to maximize material removal rate and minimize the machining error using optimal tools selected from a standard cutting tool library. Algorithms for detection and correction of local tool gouging and global tool interference will be investigated in this step.

(7). Tool path generation

Obtain optimal tool paths for fast material removal in rough cuts (low resolution) and high surface quality in fine cuts (high resolution). The set of all possible valid orientations of the tool is referred to as the configuration space. What we seek now is a tool path, which sweeps the entire delta-volume (material to be removed) without leaving the configuration space. Our approach will be to do so by interpolating between the valid orientations we have computed. Interpolation of position between orientations can be performed in a straightforward fashion using linear or spline interpolation. Interpolation of Valid tool orientations will be connected to form a valid continuous tool path. Tool path simulation is needed in this stage to avoid interference resulting from interpolation.

(8). Feed rate compensation

Feed rate compensation is needed because of variable depth of cut machining resulting from non-constant distances between offset surfaces. The feed rate compensation will be achieved incorporating a mechanistic model for cutting force (Devor et al 1980). The feed rate will be adjusted so that a uniform cutting force is maintained.

(9). Experimental work

The time efficiency of wavelet based machining with other machining algorithms will be experimentally compared.

In above steps, surface offset and multiresolution accessibility analysis are the key steps.

CHAPTER 6 CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

Wavelets based multiresolution analysis technique has been successfully used for NC tool path planning for 2.5D machining. Combined with convex hull method, an integrated rough-to-fine cut strategy was developed. The multiresolution method results in exact offsets close to the object boundary as in the contour parallel method, but approximated (coarse) curves, straight lines and circular arcs are used away from the object boundary. This results in simpler curves in rough machining, sharp corners are smoothed out thereby reducing uncut areas and larger tools can be selected.

Experimental results indicate that multiresolution tool path planning improves machining efficiency. Tool path length is reduced. In machining simulations, a reduction in machining time of up to 20% was obtained.

6.2 Scope for Future Work

Free-form pocket machining with islands will be one of the focus of our future work. Multiresolution analysis and wavelet based morphing will be used to generate tool paths in integrated rough-to-fine pocketing with islands. Outer border contour and island contour curves will be decomposed to generate two lowest resolution curves. Two set of wavelet coefficients obtained from the decomposition are interpolated, then added back to the two lowest resolution curves in each hierarchal level, two new curves are interpolated again to generate an intermediate curve as tool paths between outer boundary and island boundary. The interpolating algorithms are key in this method. Multiresolution morphing techniques may be used during the interpolation.

Another direction of future work is 3D object machining. The second generation wavelet will be used for integrated five-axis CNC machining of 3D objects. Multiresolution subdivision surface offset technique is a key for 3D machining. The visibility or accessibility information at any resolution can be determined by Visibility Maps from lower resolution and wavelet coefficients. Wavelets coefficients will also be used to obtain an optimal set of tool sizes that are needed to machine the object. The multiresolution analysis based NC tool path planning is expected to be very efficient because of the attractive linear time complexity of wavelets. It can significantly improve the productivity and the accuracy of machined parts.

REFERENCES

Austin, S.P., Jerard, P.B. and Drysdale, R.L., 1997, Comparison of Discretization Algorithms for NURBS Surfaces with Application to Numerically Controlled Machining, V29, n1, pp71-83.

Bartels, R., Beatty, J. and Barsky, B, 1987, An Introduction to Splines for use in Computer Graphics and Geometric Modeling, Morgan Kaufmann Inc.

Bedworth, D.D., Henderson, M.R., and Wolfe, P.M., 1991, Computer-Integrated Design and Manufacturing, Industrial Engineering and Management Science, McGraw-Hill, New York.

Beylkin, G., Coifman, R. and Rokhlin, V., 1991, Fast wavelet transforms and numerical algorithms I. Communications on Pure and Applied Mathematics, V44, pp.141–183.

Bobrow, J.E., 1985, Solid modelers improve NC machine tool path generation techniques, Computers in Engineering, pp. 439-444.

Chen, L.L. and Woo, T.C., 1992, “Computational geometry on the sphere with application to automated machining,” ASME Trans. J. Mech. Des. 114, pp. 285-95.

Choi, B.K. and Jerard. R.B., 1998, “Sculptured surface machining – theory and applications,” ISBN 0-412-78020-8, Dordrecht, The Netherlands: Kluwer.

Choi, B.K. and Kim, B.H., 1997 "Die-Cavity Pocketing via Cutting Simulation" Computer Aided Design, Vol. 29, No. 12, pp 837-846

Choi, B.K., Lee, C.S. Hwang, J.S. and Jun, C.S., 1988, Compound Surface Modeling and Machining, Computer Aided Design, V20, n3, pp. 127-136.

Chui, C.K., 1992, *An introduction to Wavelets*, Academic Press, Inc., Boston.

Chui, C.K. and Quak, E., 1992, *Wavelets on a bounded interval*, Numerical Methods in Application Theory, Vol. 9, pp. 53-75.

Chuang, S. H and Kao, C. Z, 1999, One-Sided Arc Approximation of B-spline Curves for Interference-Free Offsetting, Computer-Aided Design, Vol. 31, No. 2.

Chuang, S.H. and Lin W. S., 1997, Tool-Path Generation for Pockets with Freeform Curves Using Bezier Convex Hulls, Int. J. Adv. Manuf. Technol, Vol. 13, 1997, pp. 109-115.

Devor, R.E., Kline, W. A. and Zdeblick, W. J., 1980, “A Mechanistic Model of the Force System in End Milling with Application to Machining Airframe Structures,” Proc. 8th NAMRC Conference, pp. 297-303.

DeVore, R., Jawerth, B. and Lucier, B., 1992, Image compression through wavelet transform coding. IEEE Transactions on Information Theory, 38(2): 719–746, March 1992.

Dragomatz, D. and Mann, S., 1997, A classified bibliography of literature on NC milling path generation, Computer-Aided Design, Vol. 29, No. 3, pp. 239-247.

Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W., 1995, Multiresolution analysis of arbitrary meshes, SIGGRAPH, pp. 173-182.

Elber, G. and Cohen, E., 1994, "Toolpath Generation for Freeform Surface Models," Computer Aided Design, Vol. 26, No. 6, pp. 490-496.

Elber, G., Lee, I. and Kim, M., 1997, *Comparing Offset Curve Approximation Methods*, IEEE Computer Graphics and Applications," May-June 1997, pp.62-71.

Finkelstein, A. and Salesin, D.H., 1994, *Multiresolution Curves*, Computer Graphics Proceedings, Annual Conferences, pp. 261-268.

Greenwood, F., 1989, Introduction to Computer-Integrated Manufacturing, Harcourt Brace Jovanovich, FL

Held, M., Lukacs, G. and Andor,A., 1994, *Pocket machining based on contour-parallel tool paths generated by means of proximity maps*, Computer Aided Design., Vol. 26, No. 3, pp. 189-203.

Jenson, C.G. and Anderson, D.C., 1992, Accurate Tool Placement and Orientation for Finish Surface Machining, Concurrent Engineering, Ped-Vol 59, ASME, pp. 127-145.

Jenson, C.G. and Anderson, D.C., 1996, A review of numerically controlled methods for finish-sculptured-surface machining, IIE Transactions, 28, pp. 30-39.

Jenson, C. G., Mullins, S. H., Anderson, D. C., 1993, "Scallop elimination based on precise 5-axis tool placement, orientation and step-over calculations," ASME – Adv Des Automat, 65(2), pp. 535-44.

Jenson, C.G., Red, W.E., Pi, J., 2002, "Tool selection for 5-axis curvature matched machining," Computer Aided Design, 34, pp.251-66.

Kruth J. P. and Klewais, P., 1994, "Optimization and dynamic adaptation of the cutter inclination during 5-axis milling of sculptured surfaces," CIRP Ann, 43(1), pp. 443-8.

Lee, A., Sweldens, W., Schroder, P., Cowsar, L., and Dobkin, D., 1998, Maps: Multiresolution adaptive parameterization of surfaces, SIGGRAPH, pp. 95-104

Lee, Y.S. and Chang, T.C., 1995, “2-Phase approach to global tool interference avoidance in 5-axis machining,” Computer Aided Design, 27(1), pp. 715-29.

Lee, Y.S. and Chang, T.C., 1996, “Automatic cutter selection for 5-axis sculptured surface machining,” Int. J. Prod. Res., 34, pp. 997-8.

Lee, Y.S., Ji, H, 1997, *Surface interrogation and machining strip evaluation for 5-axis CNC die and mold machining,* Int J. Prod. Res, 35(1), pp. 225-52.

Loop, C., 1987, Smooth Subdivision Surface Based on Triangles, Master’s Theisi, Department of Mathematics, University of Utah.

Lounsbery, J. M., 1995, “Multiresolution analysis for surface of arbitrary topological type,” Ph.D. thesis, University of Washington.

Lounsbery, J. M., DeRose, T. and Warren, J., 1997, Multiresolution Surfaces of Arbitrary Topological Type, ACM Transactions on Graphics, Vol. 16, No 1, January, 1997, pp. 34-73.

Mallat, S., 1989, *A theory for multiresolution signal decomposition*, IEEE Trans. Pattern Anal. Machine Intell., V11, pp. 674-693.

Maekawa, T., 1996, Computation of Shortest Paths on Free-Form Parametric Surfaces, ASME Journal of Mechanical Design, V118, n4, pp. 499-508.

Marshall, S. and Griffiths, J.G., 1994, A survey of cutter path construction techniques for milling machines, INT. J. PROD. RES., 32(12), pp. 2861-77.

Oliver, J.H. and Huang, Y., 1994, Non-Constant Parameter NC tool Path Generation on Sculptured Surfaces, International Journal of Advanced Manufacturing Technology, Vol. 9, pp. 281-290.

Persson, H., 1978, *NC Machining of Arbitrary Shaped Pockets*, Computer Aided Design, Vol. 10, No. 3, pp. 169-174.

Quak, E. and Weyrich, N., 1994, Decomposition and reconstruction algorithms for spline wavelets on a bounded interval, Applied and Computational Harmonic Analysis, Vol. 1, No.3, pp. 217-231.

Rao, N., Bedi, S., Buchal, R., 1996, "Implementation of the principal-axis method for machining of complex surfaces," Int. J. Adv. Manufact Technol, 11, pp. 249-57.

Redonnet J.M., Rubio, W., Monies, F., Dessein, G., 1998, "Optimizing tool positioning for end-mill machining of free form surfaces on 5-axis machines for both semi finishing and finishing," Int. J. Adv. Manufact Technol, 16 pp. 383-91.

Sarma, R., 2000, *An Assessment of Geometric Methods in Trajectory Synthesis for Shape-Creating Manufacturing Operations*, Journal of Manufacturing Systems, Vol. 19, No. 1, pp. 59-72.

Schroder, P. and Sweldens, W., 1995, “Spherical Wavelet: Efficiently Representing Functions on the Sphere,” SIGGRAPH ’95 Proc., pp.161-72, ACM.

Shah, J.J., Sreevalsan, P. and Matthew, A., 1991, Survey of CAD/Feature-Based Process Planning and NC Programming Techniques, Computer Aided Engg. Journal, V8, n1, pp. 832-838.

Stollnitz, E.J., DeRose, T.D. and Salesin, D.H., 1995, *Wavelets for Computer Graphics: A Primer Part 1*, IEEE Computer Graphics and Applications, pp. 76-84.

Stollnitz, E.J., DeRose, T.D. and Salesin, D.H., 1995, *Wavelets for Computer Graphics: A Primer Part 2*, IEEE Computer Graphics and Applications, pp. 75-85.

Stollnitz, E., DeRose, T. and Salesin, D., 1996, Wavelets for Computer Graphics: Theory and Applications, Morgan-Kaufmann.

Suresh, K. and Yang, D.C.H., 1994, Constant Scallop Height Machining of Free Form Surfaces, ASME Journal of Engg. For Industry, V116, pp. 253-259.

Sweldens, W., 1995, The lifting scheme: A new philosophy in biorthogonal wavelet constructions, In A. F. Laine and M. Unser, editors, Wavelet Applications in Signal and Image Processing III, pp.68-79, Proc. SPIE 2569.

Tiller, W. and Hanson, E.G., 1984, Offsets of Two-Dimensional Profiles, IEEE Computer Graphics and Applications, pp. 36-46.

Toussaint, G. T., 1985, A simple linear algorithm for intersecting convex polygons, The Visual Computer, vol. 1, pp. 118-123.

Tseng, Y-J. and Joshi, S., 1994, “Recognizing Multiple Interpretations of Interacting Machining Features,” Computer Aided Design, 26(9), pp.667-688.

Vickers, G.W. and Bradley, C., 1992, *Curved Surface Machining through Circular Arc Interpolation*, Computers in Industry, Vol.19, pp. 329-337.

Wang, Y., Lee, S.L., and Toraichi, K., 1999, *Multiscale Curvature-Based Shape Representation Using B-spline Wavelets*, IEEE Transactions on Image Processing, Vol. 8, No. 11, pp. 1586-1592.

Wood, Z., Desbrun, M., Schroder, P., and Breen, D., 2000, Semi-regular mesh extraction from volumes, SIGGRAPH.