ELSEVIER

Computer-Aided Design 35 (2003) 167–178

COMPUTER-AIDED DESIGN

www.elsevier.com/locate/cad

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

Ranga Narayanaswami*, Junhua Pang

Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, IA 50011, USA

Received 5 July 2001; revised 24 November 2001; accepted 22 December 2001

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 NC machining. In this paper, 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 are used farther away from the object boundary. Experimental results indicate that wavelet-based 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. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Wavelets; Pocketing; NC machining

1. Introduction

Tool path planning 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.

In this paper, our primary concern is machining of contoured 2.5D objects. By this we refer to producing objects whose cross-sections are constant along one axis. This is similar to pocket machining an island without outer boundary walls. Two techniques are primarily used for pocketing:

- *Zigzag cutting*: First, an offset of the boundary of the pocket is intersected with a number of uniformly spaced parallel lines. The lines are then classified into segments inside and outside the pocket. Finally, the segments inside the pocket are assembled into a tool path. While the computations are straightforward [1] the demerit of this method is that it may not generate optimal tool paths when the pocket shape becomes complex. In addition, it also produces several short machining segments with frequent stops and requires an extra finishing pass to clean up the jagged boundary that results from the zigzag movements of the tool.
- *Contour-parallel cutting*: In this approach, successive offsets of the original boundary are generated. These offsets are then chained together into a single spiraling tool path that follows the contours of the pocket. 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 [2,3]. The contour parallel method is generally favored because the number of tool retractions is reduced, no final wall finishing is needed and the surface finish tends to be better because the tool moves in the same direction between consecutive passes.

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:

$$r_o(t) = r(t) \pm \mathrm{d}n(t) \quad \text{for } t \in [0,1] \tag{1}$$

where $n(t)$ is the unit normal to $r(t)$ at each point. Offset curves are thus 'parallel' to the original curve with constant

[*] Corresponding author. Tel.: +1-515-294-8730; fax: +1-515-294-3524. *E-mail address:* ranga@iastate.edu (R. Narayanaswami).

PII: S0010-4485(02)00050-7

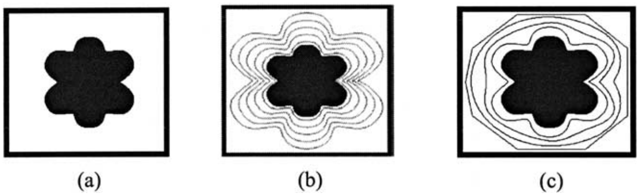

Fig. 1. Cutting sequence for 2.5D object machining. The region between the rectangular boundary and the shaded object needs to be removed.

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 [4] 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.

The theory of B-spline 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.

Fig. 1 shows an illustration where 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 provided 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.

The rest of the paper is organized as follows. The theory of wavelets is briefly reviewed in Section 2. In Section 3, the tool path for machining 2D profiles from rectangular stock based on the endpoint interpolating B-spline wavelets is derived. In Section 4, wavelet-based tool path simulations are presented. In Section 5 the wavelet-based tool path is compared with the contour parallel method. In Section 6 we suggest possible extensions to machine 3D objects based on wavelets. Conclusions are presented in Section 7.

2. Wavelets and multiresolution analysis

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

$$f(t) = \sum_{l} a_l \phi_l(t) \tag{2}$$

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:

$$f(t) = \sum_{k} \sum_{j} a_{j,k} \psi_{j,k}(t) \tag{3}$$

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.

In the following paragraphs we provide a simple introduction to wavelets using Haar wavelets. The B-spline wavelet is discussed next.

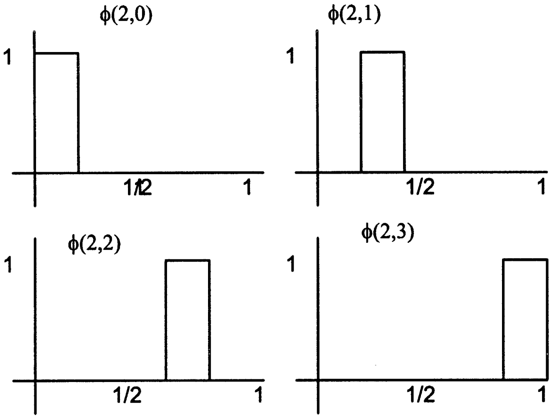

Fig. 2. The box basis for V^2.

2.1. Haar wavelets

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

$8, 6, 3, 5$

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

$7, 4$

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

$$7 + 1 = 8$$

$$7 - 1 = 6$$

$$4 + (-1) = 3$$

$$4 - (-1) = 5$$

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 $(8, 6, 3, 5)$.

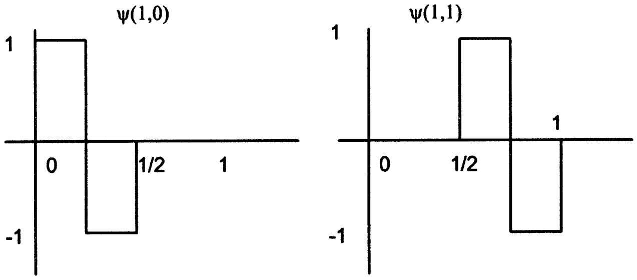

Fig. 3. The Haar wavelets for W^1.

For piecewise 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 $[0, 1/2)$ and $[1/2, 1)$. This *space* is denoted as V^1. If this is continued further, V^j will include all piecewise 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

$$\phi_i^j(x) = \phi(2^j x - i), \quad i = 0, \dots, 2^j - 1 \tag{4}$$

where

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

Fig. 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 Fig. 3 and can be specified as

$$\psi_i^j(x) = \psi(2^j x - i) \quad i = 0, \dots, 2^{j-1} \tag{5}$$

where

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

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 as $V^1 + W^1$ 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 [5]. 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. The first step is to define the scaling functions for a nested set of function spaces the second step is to define an inner product and the third step is to obtain the wavelet functions. The following paragraphs give a brief description on the generation of the cubic endpoint interpolating B-spline wavelet.

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 $i = 0, \ldots, k$ and for $r = 1, 2, 3$, let

$$N_i^0(x) := \begin{cases} 1 & \text{if } x_i \le x < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

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

(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 $N_0^3(x), \ldots, N_k^3(x)$ form a basis for the space of piecewise-polynomials of degree 3 with two continuous derivatives. For uniformly spaced cubic B-splines, $k = 2^j + 2$ and $x_4, \ldots, x_k$ 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:

$$0, 0, 0, 0, \frac{1}{2^j}, \frac{2}{2^j}, \ldots, 1 - \frac{1}{2^j}, 1, 1, 1, 1$$

At any hierarchical level j there are $2^j + 3$ control points and $2^j + 7$ knots.

The condition that the function space V^j be nested is equivalent to requiring that the scaling functions be refinable. That is, for all $j = 1, 2, \ldots$ there must exist a matrix of constants P^j such that

$$\Phi^{j-1}(x) = \Phi^j(x) P^j \tag{7}$$

where, Φ^j is a row matrix of all scaling functions at level j as shown in Eq. (8).

$$\Phi^j = \left[\phi_0^j, \phi_1^j, \ldots, \phi_{2^j+2}^j \right] \tag{8}$$

The refinement matrix P^j is of dimension $(2^j + 3) \times (2^{j-1} + 3)$. The entries of the refinement matrix can be developed using the theory of B-splines [6,7].

The second step is the choice of an inner product and the standard inner product is used for this purpose Eq. (9).

$$\langle f | g \rangle := \int_0^1 f(x) g(x) \mathrm{d}x \tag{9}$$

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:

$$\Psi^{j-1}(x) = \Phi^j(x) Q^j \tag{10}$$

Q^j is a $(2^j + 3) \times (2^{j-1})$ matrix, where Ψ^j is a row matrix of wavelet functions as in Eq. (11).

$$\Psi^{j-1} = \left[\psi_0^{j-1}, \psi_1^{j-1}, \ldots, \psi_{2^{j-1}-1}^{j-1} \right] \tag{11}$$

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

$$[\langle \Phi^{j-1} | \Psi^{j-1} \rangle] = 0 \tag{12}$$

Substituting Eq. (10) into Eq. (12) yields

$$[\langle \Phi^{j-1} | \Phi^j \rangle] Q^j = 0 \tag{13}$$

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.

The scaling functions and the wavelet functions have now been found. To use B-splines wavelets, *wavelet decomposition* (Eq. (14)) and *wavelet reconstruction* (Eq. (15)) need to be implemented.

$$[P^j | Q^j] \left[\begin{array}{c} c^{j-1} \\ \hline d^{j-1} \end{array} \right] = c^j \tag{14}$$

$$c^j = P^j c^{j-1} + Q^j d^{j-1} \tag{15}$$

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 Eq. (14) 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 Eq. (15). 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 [8–10].

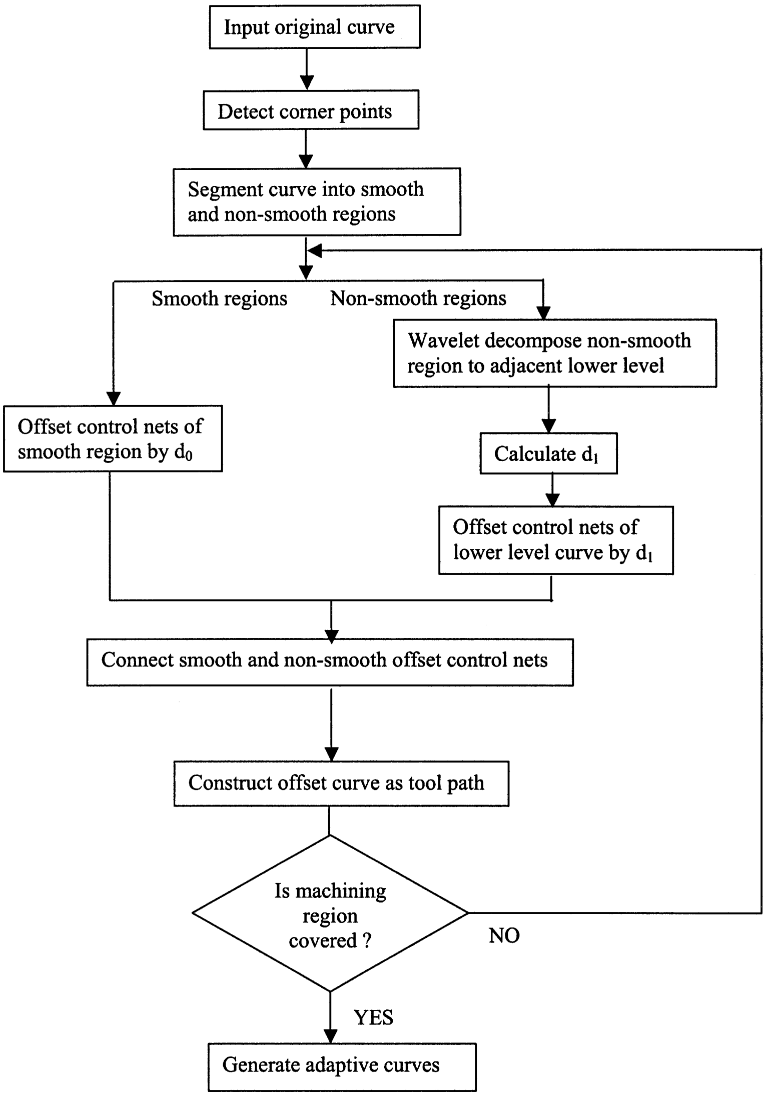
Input original curve
Detect corner points
Segment curve into smooth and non-smooth regions
Smooth regions
Non-smooth regions
Wavelet decompose non-smooth region to adjacent lower level
Offset control nets of smooth region by d0
Calculate dl
Offset control nets of lower level curve by dl
Connect smooth and non-smooth offset control nets
Construct offset curve as tool path
Is machining region covered ?
NO
YES
Generate adaptive curves

Fig. 4. Flowchart for wavelet-based tool path curve generation.

3. 2D profile machining

In this section, 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 [11]:

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.

The overall algorithm for wavelet-based tool path generation is shown in Fig. 4.

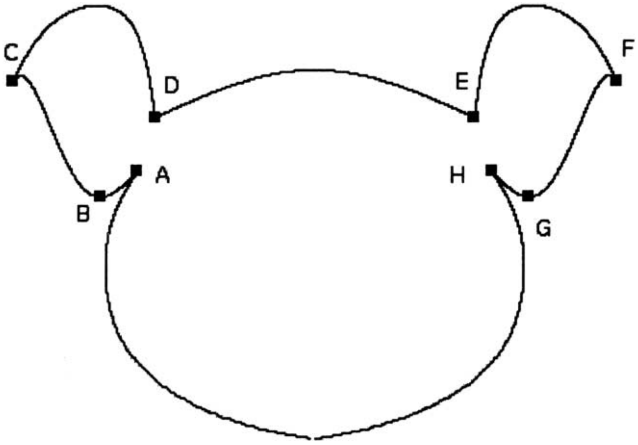

Fig. 5. Segmentation of object contour based on corner point detection.

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. [12] to detect the corner points. Curvature functions of a B-spline curve at different scales j can be estimated using

$$k(u,j) = \frac{W_1x(u,j)W_2y(u,j) - W_1y(u,j)W_2x(u,j)}{(W_1x(u,j)^2 + W_1y(u,j)^2)^{3/2}} \tag{16}$$

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

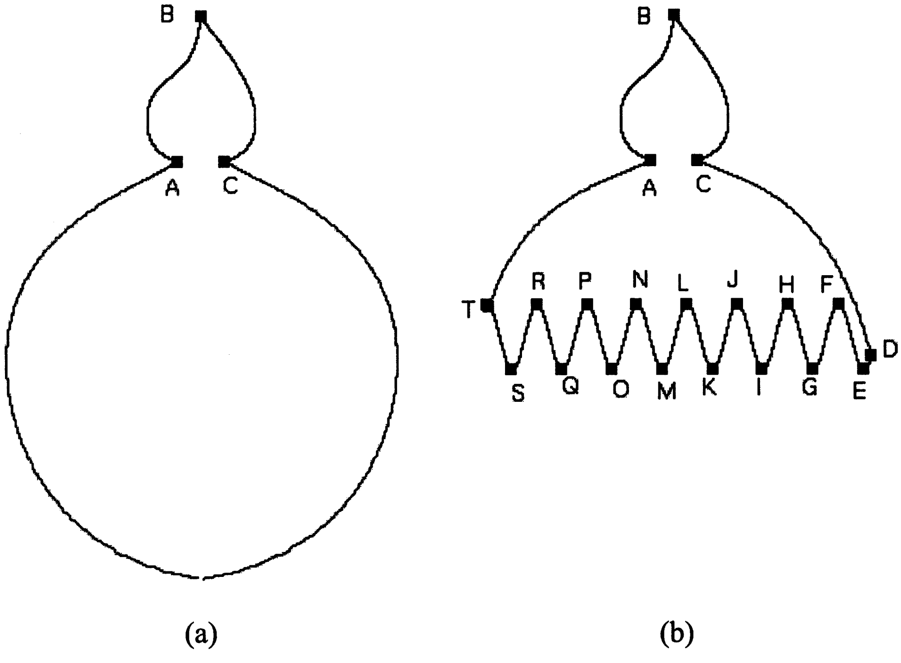

Fig. 6. Relative smooth and non-smooth regions on an object contour.

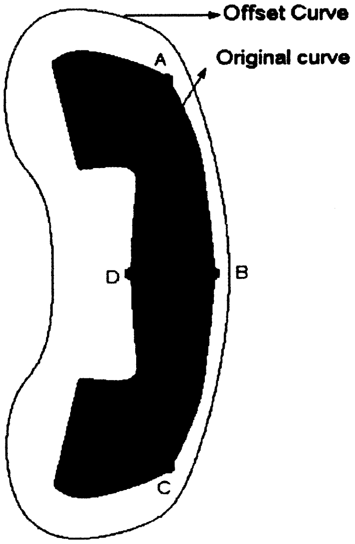

Fig. 7. Generation of an offset contour for a telephone shaped object. Different offset distances to smooth regions and non-smooth regions are applied. Smooth region (curve ABC) is offset by distance d_0. Non-smooth region (curve CDA) is offset by distance d_1 ($d_1 > d_0$).

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 Fig. 5. 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. In Fig. 6 we compare two objects with a similar feature. In Fig. 6(a) segment ABC is identified as a non-smooth region. However, in Fig. 6(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 Fig. 6(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 Eq. (14). 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 [13] which compares well with other offset techniques [14].

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 $\epsilon$;
  While deviation $> \epsilon$ 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 $v$;
```

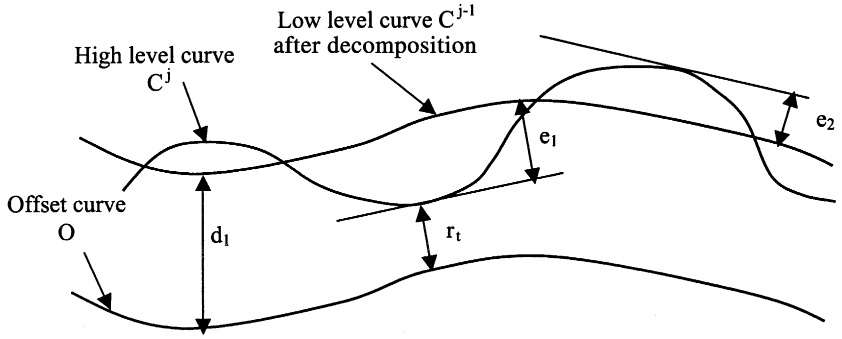

Fig. 8. Offset distance d_1 in non-smooth region.

```
    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.

Fig. 7 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.

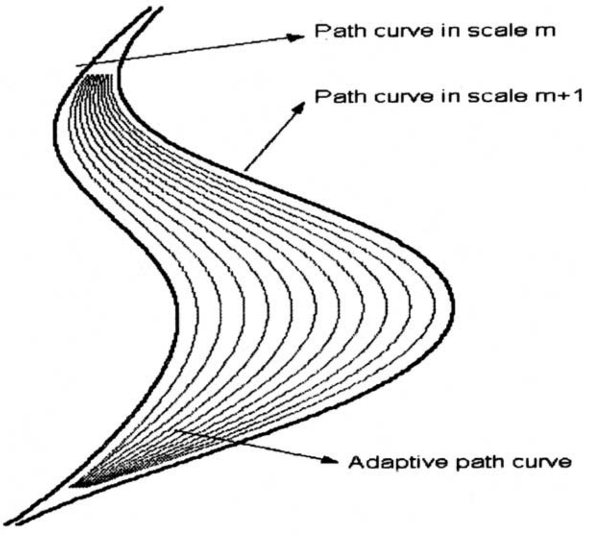

Fig. 9. Adaptive tool path curves. The adaptive tool path curve is used to create a more uniform distance between successive tool paths.

The value for d_1 may be computed from the stored wavelet coefficients. Fig. 8 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, 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 [15]. If E^j is defined as a column vector

$$E^j = M^j Q^j D^{j-1} \tag{17}$$

where, M^j is the B-spline to Bezier conversion matrix, Q^j is the wavelet decomposition matrix as in Eq. (14) 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 [8].

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

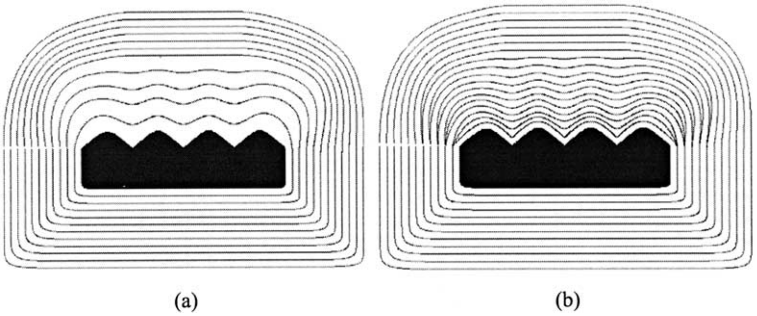

Fig. 10. Wavelet-based tool path for mountain shaped object. (a) Basic tool path curves are generated; (b) adaptive curves are added.

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:

$$c^{m+t} = P^{m+1}c^m + tQ^{m+1}d^m \tag{18}$$

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 Eq. (14) and c^{m+t} is the newly computed set of control points.

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 Fig. 8. Therefore, the number of adaptive curves n that is needed between two adjacent scale curves can be calculated as:

$$n = (e_1 + e_2)/r_t \tag{19}$$

Fig. 9 shows an illustration of adaptive curves based on this technique.

4. 2D 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. Fig. 10(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 Fig. 10(b) the adaptive curves are added. Fig. 11 shows the basic and adaptive tool path curves for a telephone shaped object.

An airplane shaped object and a rabbit shaped object are also shown to illustrate the wavelet-based tool path. Figs. 12 and 13 show the tool path. If necessary, the outer boundaries can be made linear.

5. 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

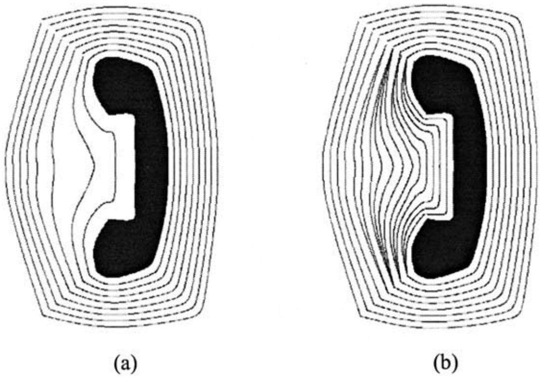

Fig. 11. Wavelet-based tool path for telephone shaped object. (a) Basic tool path curves are generated; (b) adaptive curves are added.

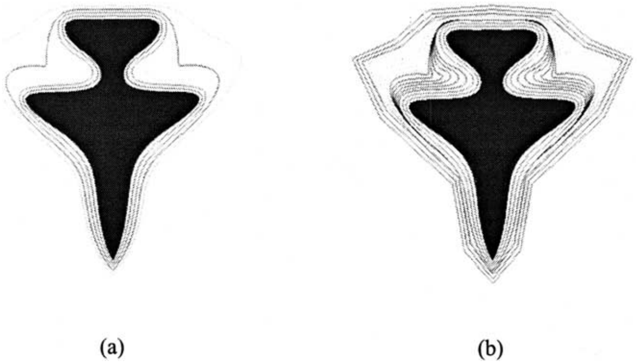

Fig. 12. Tool path for an airplane shaped object. (a) Basic tool path curves; (b) adaptive curves.

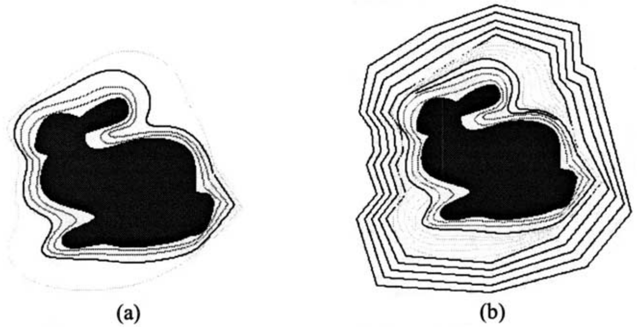

Fig. 13. Tool path for a rabbit shaped object. (a) Basic tool path curves; (b) adaptive curves.

comparison, the machining areas are assumed equal and the machining time is compared using the ProCAM TekSoft-2D machining simulation software. In Fig. 14(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 Fig. 14(b) the wavelet-based tool path is shown for the mountain shaped object. 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 1, the tool path length is reduced from 752.37 in the contour parallel method to 510.19 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 Fig. 15 the contour parallel and wavelet-based tool path are compared for the telephone shaped object. As shown in Table 1, in this case the tool path length is reduced from 914.38 to 835.47. This is a decrease of 9%. The machining time is reduced from 77.52 to 71.87 min corresponding to a decrease of 7%.

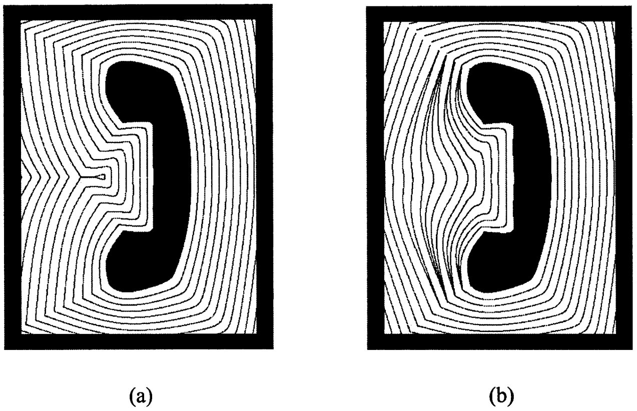

Fig. 15. Machining simulation comparison between contour parallel (a) and wavelet-based method (b).

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 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 [16] 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.

Smoother tool paths improve machining efficiency by avoiding uncut areas as well. Fig. 14(a) and (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

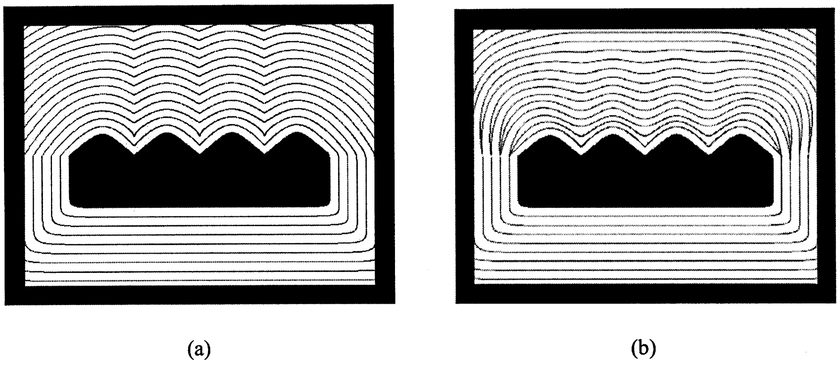

Fig. 14. Machining simulation comparison between contour parallel (a) and wavelet-based method (b).

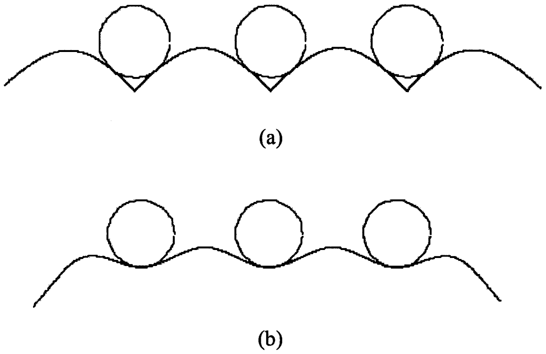

Fig. 16. Uncut regions that exist in contour parallel method (a) may be reduced in wavelet-based method (b).

machining as compared to contour parallel machining (Fig. 16).

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.

6. Extension to 3D objects

The 2D method can conceivably be used to machine 3D objects in slices. The 3D part may be first segmented into slices in a preferred direction. The intersection curves of the slicing planes with the object model may then be used to generate 2D offset curves in each slice. Based on the slice width a layered or step pattern may be created as an artifact that would require a smoothing cut would then be required.

Other wavelets, such as, the Haar and the subdivision wavelet may offer alternative approaches for 3D machining. The Haar wavelet can provide techniques to plan rough cut machining on three axis machines and the subdivision wavelet can be useful for rough and finish cuts on four and five axis machines. Using the subdivision wavelet, a complex surface can be represented at a very high-resolution with very small triangular facets [17]. At lower resolutions fewer triangles with a larger size are used. Fig. 16 shows an example of the subdivision wavelet. Fig. 17(a) shows the high-resolution polyhedron and Fig. 17(b) shows a lower resolution approximation and finally Fig. 17(c) shows the lowest resolution approximation. In terms of NC machining, the larger size triangles after offset modification can be used for the rough-cuts. In the finish machining stage, smaller triangular facets need to be machined. The complete capability of a four or five axis machine can be used effectively this way. We believe that the subdivision wavelet technique is a promising method for increasing the cutting efficiency of multi-axis machines and improving the accuracy of machined parts.

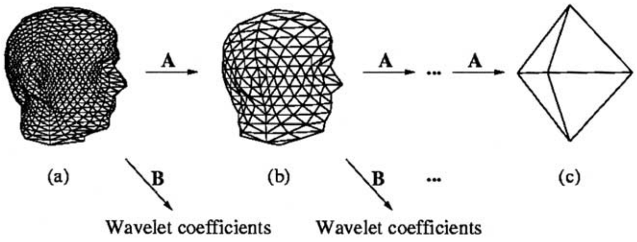

Fig. 17. 3D object representation at various scales using the subdivison wavelet.

7. Conclusion and future work

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.

Future work is focused on improved object segmentation methods and extensions to 3D object machining. The subdivision wavelet will be used for integrated four and five axis CNC machining of 3D objects.

Table 1
Comparison of wavelet-based method and contour parallel method. (Tool radius is 0.25 in., 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 4 in. × 6 in. and for the telephone shaped object is 5 in. × 7 in.)

	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

Acknowledgements

Support from NSF DMII 9970083 is gratefully acknowledged.

References

[1] Lee YS, Chang TC. Application of computational geometry in optimizing 2.5D and 3D NC surface machining. Comput Ind 1995;26(1):41–59.

[2] Persson H. NC machining of arbitrary shaped pockets. Comput Aid Des 1978;10(3):169–74.

[3] Held M, Lukacs G, Andor A. Pocket machining based on contour-parallel tool paths generated by means of proximity maps. Comput Aid Des 1994;26(3):189–203.

[4] Vickers GW, Bradley C. Curved surface machining through circular arc interpolation. Comput Ind 1992;19:329–37.

[5] Chui CK. An introduction to wavelets. Boston: Academic Press, 1992.

[6] Chui CK, Quak E. Wavelets on a bounded interval. Num Meth Appl Theory 1992;9:53–75.

[7] Quak E, Weyrich N. Decomposition and reconstruction algorithms for spline wavelets on a bounded interval. Appl Comput Harmonic Anal 1994;1(3):217–31.

[8] Finkelstein A, Salesin DH. Multiresolution curves. Computer Graphics Proceedings, Annual Conferences, 1994. p. 261–8.

[9] Stollnitz EJ, DeRose TD, Salesin DH. Wavelets for computer graphics: a primer part 1. IEEE Comput Graph Appl 1995:76–84.

[10] Stollnitz EJ, DeRose TD, Salesin DH. Wavelets for computer graphics: a primer part 2. IEEE Comput Graph Appl 1995:75–85.

[11] Narayanaswami R, Pang J. Toolpath Planning Based on B-spline wavelets. ASME Design Automation Conference, Pittsburgh, PA, USA, 2001.

[12] Wang Y, Lee SL, Toraichi K. Multiscale curvature-based shape representation using B-spline wavelets. IEEE Trans Image Process 1999;8(11):1586–92.

[13] Tiller W, Hanson EG. Offsets of two-dimensional profiles. IEEE Comput Graph Appl 1984:36–46.

[14] Elber G, Lee I, Kim M. Comparing offset curve approximation methods. IEEE Comput Graph Appl 1997:62–71.

[15] Bartels R, Beatty J, Barsky B. An introduction to splines for use in computer graphics and geometric modeling. Los Altos, CA: Morgan Kaufmann, 1987.

[16] Elber G, Cohen E. Toolpath generation for freeform surface models. Comput Aid Des 1994;26(6):490–6.

[17] Lounsbery M, DeRose T, Warren J. Multiresolution surfaces of arbitrary topological type. ACM Trans Graph 1997;16(1):34–73.

Ranga Narayanaswami is an Assistant Professor in the department of Industrial and Manufacturing Systems Engineering at Iowa State University. His research interests are in Computer aided design, geometric modeling, computer graphics and its applications in manufacturing. He holds a PhD in mechanical engineering from the University of California at Berkeley and a BS from the Indian Institute of Technology at Madras.

Junhua Pang is a graduate student in the department of Industrial and Manufacturing Systems Engineering at Iowa State University. His research interests are in computer-aided design and manufacturing. He holds an MS from Beijing Institute of Technology and a BS from the Zheijiang University, China both in mechanical engineering.