

2000

Transform-based surface analysis and representation for CAD models

Lei Sun

Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Computer Sciences Commons](#), and the [Mechanical Engineering Commons](#)

Recommended Citation

Sun, Lei, "Transform-based surface analysis and representation for CAD models " (2000). *Retrospective Theses and Dissertations*. 12362.
<https://lib.dr.iastate.edu/rtd/12362>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

Transform-based surface analysis and representation for CAD models

by

Lei Sun

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Mechanical Engineering
Major Professor: Abir Z. Qamhiyah

Iowa State University

Ames, Iowa

2000

Copyright © Lei Sun, 2000. All right reserved.

UMI Number: 9990489

UMI[®]

UMI Microform 9990489

Copyright 2001 by Bell & Howell Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

Bell & Howell 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
Lei Sun
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

Signature was redacted for privacy.

For the Graduate College

TABLE OF CONTENTS

LIST OF FIGURES	vi
LIST OF TABLES	xi
CHAPTER 1. INTRODUCTION	1
1.1 Motivation	1
1.2 Literature Review	3
1.2.1 Definition of Features	4
1.2.2 Representation of Solid Models	7
1.2.3 Feature Recognition	12
1.2.4 Shape Analysis and Description	17
1.3 Research Objective	21
1.4 Overview of Chapters	23
CHAPTER 2. WAVELET DECOMPOSITION	24
2.1 Wavelet in Comparison with Windowed Fourier Transforms	24
2.2 Fundamentals of the Wavelet Transforms	27
2.2.1 Continuous Wavelet Transform	27
2.2.2 Discrete Wavelet Transforms	28
2.3 Redundant Discrete Wavelet Transform Frames	28
2.4 Orthonormal Wavelet Bases	29
2.4.1 Battle-Lemarie Wavelets	31

2.4.2	Harr Wavelet Bases	33
2.4.3	Daubechies Compactly Supported Wavelet Bases	33
2.5	The Pyramid Algorithm for DWT Decomposition	35
2.6	Three Dimensional Extension of Wavelet Bases	40
CHAPTER 3. WAVELET-BASED SURFACE ANALYSIS FOR FORM FEATURE DETERMINATION		45
3.1	Methodology	45
3.2	Detailed Description of the Surface Ranking Procedure	51
3.3	Discussion of Experimental Results	71
3.3.1	Studying the Effect of Using Different Wavelet Bases and Vanishing Moments on the Representation	71
3.3.2	Studying the Effect of Level of Wavelet Transform on the Representation	73
3.4	Summary and Conclusion	73
CHAPTER 4. PARAMETRIC FACE CODING FOR INVARIANT FEATURE REPRESENTATION		76
4.1	Introduction	76
4.2	Previous Work on Face Representation	78
4.2.1	Curvature-Based Description of Geometric Characteristics of Surfaces	78
4.2.2	Contour-Oriented Shape Representation and Description	80
4.3	Topological and Geometrical Entities in ACIS Modeling Kernel	82
4.4	Surface Differential Geometry	84

4.4.1 Surface Normal Curvature	84
4.4.2 Calculation of Surface Curvature	86
4.5 The Parametric Face Coding Procedure	89
4.5.1 Wavelet-Based Face Region Coding	91
4.5.2 Face Loop Coding by Wavelet-Based Zero-Crossing	100
4.6 An Example	105
4.7 Summary	141
CHAPTER 5 CONCLUSIONS AND FUTURE WORK	142
6.1 Summary and Conclusions	142
6.2 Future Research Recommendations	145
BIBLIOGRAPHY	147

LIST OF FIGURES

Figure 1.1: A labeled block solid	9
Figure 1.2: Winged-edge representation of the block in Figure 1.1	10
Figure 1.3: Face-edge graph of the block in Figure 1.1	10
Figure 2.1: Windowed Fourier transforms in time frequency plane	25
Figure 2.2: Wavelet transforms in time frequency plane	26
Figure 2.3: The scaling function $\phi(t)$ of DAUB4 wavelet transform	37
Figure 2.4: The mother wavelet $\phi(t)$ DAUB4 wavelet transform	37
Figure 2.5: The scaling function $\phi(t)$ of DAUB8 wavelet transform	38
Figure 2.6: The mother wavelet $\phi(t)$ DAUB8 wavelet transform	38
Figure 2.7: The first level pyramid decomposition with DAUB4 for a vector of size 8	40
Figure 2.8: The first level 3D basis construction of Harr wavelets	42
Figure 2.9: Quadrature mirror filter pyramid to implement the 3D-wavelet transform	43
Figure 2.10: Volume data decomposed into eight blocks by 3D wavelets	44
Figure 3.1: Kim's part from NIST design repository	46
Figure 3.2: Surface singularity patterns in a binary representation	46
Figure 3.3: Top view of the digitized model in Figure 3.1	47
Figure 3.4: One level wavelet decomposition of top view image in Figure 3.3	48
Figure 3.5: A surface with its loops and patches	51
Figure 3.6: A snapshot of GUI interface for surface analysis research	52
Figure 3.7: An ACIS part with free-form features on its top surfaces	53

Figure 3.8: Conjugate mirror filter pyramid to implement the 3D-wavelet Transform	55
Figure 3.9: Replacing wavelet approximation coefficients $A_{j,\dots} f$ with zeros	57
Figure 3.10: Surface levels through the application of Harr DWTs	59
Figure 3.11: Surface levels through the application of Daubechies DWTs with vanishing four moment	60
Figure 3.12: Numbered surfaces for the part	61
Figure 3.13: Level distribution for surfaces in the part by Harr wavelets	64
Figure 3.14: Level distribution for surfaces in the part by Daubechies wavelets	65
Figure 3.15: Graph structures for surface clusters through the application of Daubechies wavelets	68
Figure 3.16: Second classification of surfaces in Figure 3.12 with additional manufacturing information	70
Figure 3.17: A cell's six neighborhoods and its different locations	72
Figure 4.1: Uniformly sampling $(CWT_{\sigma} x_r)(u,s)$ and $(CWT_{\sigma} x)(u,s)$ producing different values	77
Figure 4.2: Topology and geometry information in ACIS solid molder	83
Figure 4.3: Surface mapping	84
Figure 4.4: Curve curvature definition	85
Figure 4.5: A normal curvature of a surface S at a point P	86
Figure 4.6: Parametric surface patches	86
Figure 4.7: An ACIS wiggle model	90
Figure 4.8: Curvature map for the top spline surface in Figure 4.7	93
Figure 4.9: The first level decomposition of Figure 4.8	93

Figure 4.10: The second level decomposition of Figure 4.8	94
Figure 4.11: The third level decomposition of Figure 4.8	94
Figure 4.12: Three-level DAUB4 wavelet decomposition in Matlab Wavelet Toolbox	95
Figure 4.13: Location (d, θ) of a point P in relative to the point M with the maximum coefficient	97
Figure 4.14: Three-level curvature representation from detail components of wavelet decomposition for the top spline face in Figure 4.7	99
Figure 4.15: Radial distance function for the face loop	101
Figure 4.16: Radial distance function for the loop of the top spline face in Figure 4.7	101
Figure 4.17: Three-level wavelet decomposition of radial distance function in Figure 4.16	103
Figure 4.18: Level-1 wavelet zero-crossing representation of spline face loop in Figure 4.7	104
Figure 4.19: Level-2 wavelet zero-crossing representation of spline face loop in Figure 4.7	104
Figure 4.20: Level-3 wavelet zero-crossing representation of spline face loop in Figure 4.7	105
Figure 4.21: An ACIS part with free-form features on its top surfaces	107
Figure 4.22: Map of maximum principal curvature of spline surface-5 in Figure 4.21	108
Figure 4.23: The level-1 decomposition of curvature map in Figure 4.22	108
Figure 4.24: The level-2 decomposition of curvature map in Figure 4.22	109
Figure 4.25: The level-3 decomposition of curvature map in Figure 4.22	109
Figure 4.26: Three-level curvature representation from detail components of wavelet decomposition for the spline face-5 in Figure 4.21	111

Figure 4.27: Map of principal curvature of spline surface-8 in Figure 4.21	112
Figure 4.28: The level-1 decomposition of curvature map in Figure 4.27	112
Figure 4.29: The level-2 decomposition of curvature map in Figure 4.27	113
Figure 4.30: The level-3 decomposition of curvature map in Figure 4.27	113
Figure 4.31: Three-level curvature representation from detail components of wavelet decomposition for the spline face-8 in Figure 4.21	115
Figure 4.32: Radial distance function for the loop of the conical face-1 in Figure 4.21	116
Figure 4.33: Three-level wavelet decomposition of radial distance function in Figure 4.32	116
Figure 4.34: Three-level wavelet zero-crossing representation of the loop of conical face-1 in Figure 4.21	118
Figure 4.35: Radial distance function for the loop of conical face-2 in Figure 4.21	119
Figure 4.36: Three-level wavelet decomposition of radial distance function in Figure 4.35	119
Figure 4.37: Three-level wavelet zero-crossing representation of the loop of conical face-2 in Figure 4.21	121
Figure 4.38: Radial distance function for the loop of face-3 in Figure 4.21	122
Figure 4.39: Three-level wavelet decomposition of radial distance function in Figure 4.38	122
Figure 4.40: Three-level wavelet zero-crossing representation of the loop of conical face-3 in Figure 4.21	124
Figure 4.41: Radial distance function for the loop of the spline face-5 in Figure 4.21	125
Figure 4.42: Three-level wavelet decomposition of radial distance function in Figure 4.41	125
Figure 4.43: Three-level wavelet zero-crossing representation of the loop of	

spline face-5 in Figure 4.21	127
Figure 4.44: Radial distance for the loop of the cylindrical face-6 in Figure 4.21	128
Figure 4.45: Three-level wavelet decomposition of radial distance function in Figure 4.44	128
Figure 4.46: Three-level wavelet zero-crossing representation of the loop of cylindrical face-6 in Figure 4.21	130
Figure 4.47: Radial distance function for the loop of the spline face-8 in Figure 4.21	131
Figure 4.48: Three-level wavelet decomposition of radial distance function in Figure 4.47	131
Figure 4.49: Three-level wavelet zero-crossing representation of the loop of spline face-8 in Figure 4.21	133
Figure 4.50: Radial distance function for the loop of the face-9 in Figure 4.21	134
Figure 4.51: Three-level wavelet decomposition of radial distance function in Figure 4.50	134
Figure 4.52: Three-level wavelet zero-crossing representation of the loop of the face-9 in Figure 4.21	136
Figure 4.53: Radial distance function for the loop of the face-10 in Figure 4.21	137
Figure 4.54: Three-level wavelet decomposition of radial distance function in Figure 4.53	137
Figure 4.55: Three-level wavelet zero-crossing representation of the loop of the face-10 in Figure 4.21	139
Figure 4.56: The graphical representation of model in Figure 4.21 with face region and loop coding.	140

LIST OF TABLES

Table 2.1: The low filter coefficients $h[n]$ for compact supported Daubechies wavelets	36
Table 3.1: Options for wavelet coefficient manipulation before reconstruction	50
Table 3.2: Surface-level representation by Harr wavelets	58
Table 3.3: Surface-level representation by Daubechies wavelets with vanishing moment	61
Table 3.4: Harr surface level representation thresholded by 30%	63
Table 3.5: Daubechies surfaces level representation thresholded by 30%	63
Table 3.6: Updated level matrix by Daubechies wavelets after Surfaces 5, 6, and 8 are identified	67
Table 3.7: Updated level matrix by Daubechies wavelets after Surfaces 1, 2, and 6 are identified	67
Table 3.8: Additional manufacturing information for form clusters in Figure 3.15	69
Table 4.1: Multi-level coding for the top spline face in Figure 4.7	98
Table 4.2: Multi-level coding for the spline face loop in Figure 4.7	103
Table 4.3: Multi-level coding for the spline surface-5 in Figure 4.21	110
Table 4.4: Multi-level coding for the spline surface-8 in Figure 4.20	114
Table 4.5: Multi-level coding for the loop of conical face-1 in Figure 4.21	117
Table 4.6: Multi-level coding for the loop of conical face-2 in Figure 4.21	120
Table 4.7: Multi-level coding for the loop of face-3 in Figure 4.21	123
Table 4.8: Multi-level coding for the loop of spline face-5 in Figure 4.21	126
Table 4.9: Multi-level coding for the loop of face-6 in Figure 4.21	129
Table 4.10: Multi-level coding for the loop of spline face-8 in Figure 4.21	132

Table 4.11: Multi-level coding for the loop of the face-9 in Figure 4.21	135
Table 4.12: Multi-level coding for the loop of the face-10 in Figure 4.21	138

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my major professor Abir Z. Qamhiyah for her invaluable and constant help in my research and in the writing of the dissertation. I am sincerely grateful for her guidance and continuous encouragement throughout this work.

I would also like to take this opportunity to thank Dr. Les Miller, Dr. Carolina Cruz-Neira, Dr. Donald R. Flugrad, Dr. James E. Bernard, and Dr. James H. Oliver for serving on my graduate committee.

Finally, special thanks to my wife, Wenqin, for her support and continuous encouragement all these years.

CHAPTER 1

INTRODUCTION

1.1 Motivation

In most Computer-Aided Design (CAD) systems, the topological and geometrical information in a CAD model is usually represented by the edge-based data structure [Mantyla 1988, LaCourse 1995, Higashi *et al.* 1997]. The level of information in the model is very low. With the emergence of concurrent engineering, such issues as product design, manufacturing, and process planning are considered simultaneously at the design stage. The need for the development of high-level models for completely documenting the geometry of a product and supporting manufacturing applications, such as automating the verification of a design for manufacturing (DFM) rules and generating process plans, becomes apparent. To achieve this goal, knowledge of the overall geometry of the product and the location of various geometric features on the product model are required. Feature-based modeling, as a link between design and manufacturing, generalizes these high-level product models.

Features can be informally defined as “generic shapes or other characteristics of a part with which engineers can associate knowledge useful for reasoning about the part (Shah and Mantyla, (1995)).” With this definition, most graph-based feature recognition approaches usually evaluate the entire low-level data structure and map it into a high-level structure. These approaches can successfully identify features with planar surfaces or cylindrical

surfaces. However, these techniques are experiencing problems with intersecting features and features with free-form surfaces.

To expand the domain of feature extraction to include products with non-planar surfaces, this dissertation develops a generic mathematical framework for feature shape descriptions. A solid model can be digitized by uniform cells and represented in a binary three-dimensional array. A spatially digitized solid model can then be transformed into a “frequency” representation that characterizes the pattern of its multi-scale details. Based upon the binary representation, both spatial and frequency properties can be extracted and used for product representation.

In most B-Rep based feature studies, a manufacturing feature is generally defined as a set of connected faces that can be simultaneously generated by a common manufacturing process. Such features could vary widely in shape due to the occurrence of interacting features. Instead of modeling multiple faces, this research proposes modeling the manufacturability of each single face at a time. A set of connected faces sharing a common manufacturing tool are then grouped together as a manufacturing feature. This dissertation focuses on the development of a mathematical framework for the extraction and classification of form features. A face is selected as the unit for feature representation, i.e. “feature primitive.” A face code is obtained from the digitized solid model through the application of 3D discrete wavelet transforms. Form features are then extracted by clustering the coded faces.

In this research, three-dimensional discrete wavelet transform (3D DWT) is used for the shape analysis of the digitized CAD models. 3D DWT can decompose a volume into a set of orthogonal, spatially oriented representations. In the binary representation, form features can

be interpreted as singularities in a product's "shape signal." A singularity is an abrupt change in a shape signal. A surface is the boundary of transition between the inside and outside of the product. As a result, a surface is presented as a sharp contrast in the binary representation. A form feature can therefore be characterized by the singularity pattern of its surfaces. When 3D DWT is applied to the 3D binary matrix, it can detect the sharp contrast caused by a feature's surfaces and assign different detail coefficients according to the surfaces' curvature patterns.

1.2 Literature Review

Features are widely regarded as key components for the development of the downstream applications, such as the automation of manufacturability analysis. Features do not have a formal mathematical definition. A variety of definitions have been proposed with respect to the engineering functionality for a specific application domain. A survey of the proposed definitions will be first presented in this section.

An understanding of the efforts in feature recognition requires an introduction to the internal representation of objects in CAD packages. Therefore, a survey of CAD models and their representations would precede that of feature recognition.

There are essentially two techniques to obtain the feature information, namely Feature-Based Modeling and Feature Recognition. With the first approach, a designer models the part directly in terms of predefined manufacturing features, so the form features required to analyze manufacturability are readily available in the CAD databases. However, in the design phase, the designer creates desired geometry in terms of function and shape instead of

manufacturing operations. Also, there are often multiple ways of interpreting a part in terms of manufacturing features. The second approach allows the designer to model a product using an existing CAD package and automatically detects features from the geometric models. Feature recognition has an advantage over feature-based modeling in that it utilizes the existing CAD databases.

In this dissertation, a wavelet-based surface analysis strategy is used for form feature extraction and coding. Therefore, this section is concluded with the presentations of the background on shape analysis and representation in both computer vision and computer aided geometric design research fields.

1.2.1 Definition of Features

The feature research community has not reached an agreement on a canonical definition for features. Researchers used to give their own distinct descriptions for features in a general way in respect to engineering meaning or significance. In order to provide a formal quantifiable measure of the geometric feature complexity, Little et al. (1998) defined a feature complexity index (FCI). FCI is an alphanumeric symbol and has three parts designed respectively for feature geometry, number of machining approach directions, and the sum of face cycles found in face-edge graph.

Various attributes need to be encapsulated in the feature model so that these attributes can be easily retrieved for the automatic process planning and manufacturing analysis purposes. In the work of Elmaragpy et al. (1994), the following attributes are listed:

1. **Geometric and topological attributes: geometric entities, primitives and Boolean operations on primitives;**
2. **Precision attributes: geometric and dimensional tolerances, datum and surface finishes;**
3. **Assembly attributes: the relations between design attributes and datum hierarchies;**
4. **Functional attributes: those relating to application specific functions and operations;**
5. **Manufacturing attributes: domain specific information about capabilities of manufacturing processes.**

With the above attribute requirements for the feature definition, several generic types of features independent of application domains can be found in the literature [Shah and MTMntyITM, 1995]. Most notions of these features can be covered by the following categories [Shah and Rogers, 1988]:

- **Form features (nominal geometry):**
 - functional
 - aesthetic
 - assembly aids
- **Material features (material composition and condition):**
 - properties/specifications
 - treatment, applied to materials and surfaces
- **Precision features (information related to part performance and operation) :**
 - Performance parameters
 - Operating variables

- Design constraints.
- Technological features (information related to part performance and operation):
 - Performance parameters
 - Operating variables
 - Design constraints.

The feature concept has its origin with the computer-aided process planning of machined parts [Subrahmanyam and Wozny, 1995]. A number of works have focused on machining features used in computer-aided process planning systems. According to Henderson (1994), a machining feature corresponds to a set of adjacent surfaces that commonly occur on the machined parts. Wu and Liu (1996) considered a machined feature as a set of similar composing faces satisfying a group of constraints, reproducible by a common cutter. Dong and Vijayan (1995) defined a machining feature as an optimal volume to be removed from the blank material within a given machine setup. The number of setups and the cost to manufacture the component are therefore minimized.

Recently there has been interest in other types of manufacturing processes, such as sheet metal forming [Chuang and Huang, 1995], casting [Stefano, 1997] and injection molding [Fu et al., 1999].

A number of research attempts have been made to define and classify general manufacturing features. In the work of Shah and Mantyla (1995), the term manufacturing feature was defined as “a collection of related geometric elements which as a whole correspond to a particular manufacturing method or process or can be used to reason about the suitable manufacturing methods or processes for creating the geometry.” In 1999, Stage,

Roberts and Henderson reported an automated Manufacturability Evaluation (ME) system at Arizona State University. The authors developed an independent definition of feature called resource-based manufacturing feature. This new type of feature incorporated available factory resources and tool accessibility information. The feature extraction algorithm selected a surface with its machine, tool and setup information as feature primitive. It then grouped feature primitives into manufacturing features in a clustering process that uses heuristic constraints and a user-defined evaluation objective.

1.2.2 Representation of Solid Models

The field of solid modeling covers a wide area of activity directed toward the representation and manipulation of 3D surfaces and volumes. In a CAD system, a physical solid object has a complete, valid, and unambiguous mathematical representation known as the solid model [Hoffmann 1989, Mantyla 1988 and Mortenson 1995]. The solid modeling technique is used to model a close and bounded object and enables spatial points to be located relative to the object (spatial addressability) [Zeid, 1991]. There are three popular classes of schemes for the representation of Solid Models and corresponding internal data structures and algorithms available in CAD.

B-Rep is the most commonly used CAD model. In the B-Rep scheme, a solid model is represented by its shells, surfaces, loops, edges and vertices. A shell is defined as a connected set of surfaces. The boundary of a surface is defined by a cyclic set of edges referred to as the outer loop. An inner loop on the other hand consists of a cyclic set of edges contained within the boundaries of a surface.

B-Rep modelers store the evaluated geometry as a redundant hierarchy of topological entities with pointers to geometric entities. Euler equations are used to ensure the validity of objects. A B-Rep model uses a data structure that represents the geometry and topology (spatial relations) of its bounding faces. One of the most popular B-Rep structures is the winged-edge representation and its variations [Weiler 1985 and Mantyla 1988]. As an example, the winged-edge representation of the block presented in Figure 1.1 is given in Figure 1.2.

B-Rep can be viewed as a directed-labeled graph, which can be used to represent relationships between faces, edges and vertices. Most feature recognition algorithms work on the B-Rep model. The spatial information available in the B-Rep model, that is, B-Rep primitives and their topology, are used directly to obtain characteristic functions. These approaches usually used face-edge graphs stored in the solid model. For the block in Figure 1.1, its face-edge graph is shown in Figure 1.3, where a node denotes a face and an arc denotes an adjacent relationship between two nodes/faces; an attribute value 0 or 1 can be assigned on each arc to denote that the intersection angle of the two adjacent faces is concave or convex. Graph matching (by purely topological information), syntactic recognition (by geometry as well as topology), and rule-based algorithm belong to this category.

Decomposition approaches represent a solid by a set of basic non-overlapping primitives glued together. Based on the types of basic primitives and the way the combination of the basic primitives is stored, decomposition models can be further classified into one of three sub-categories: (1) Exhaustive enumeration, (2) Space subdivision, and (3) Cell decomposition.

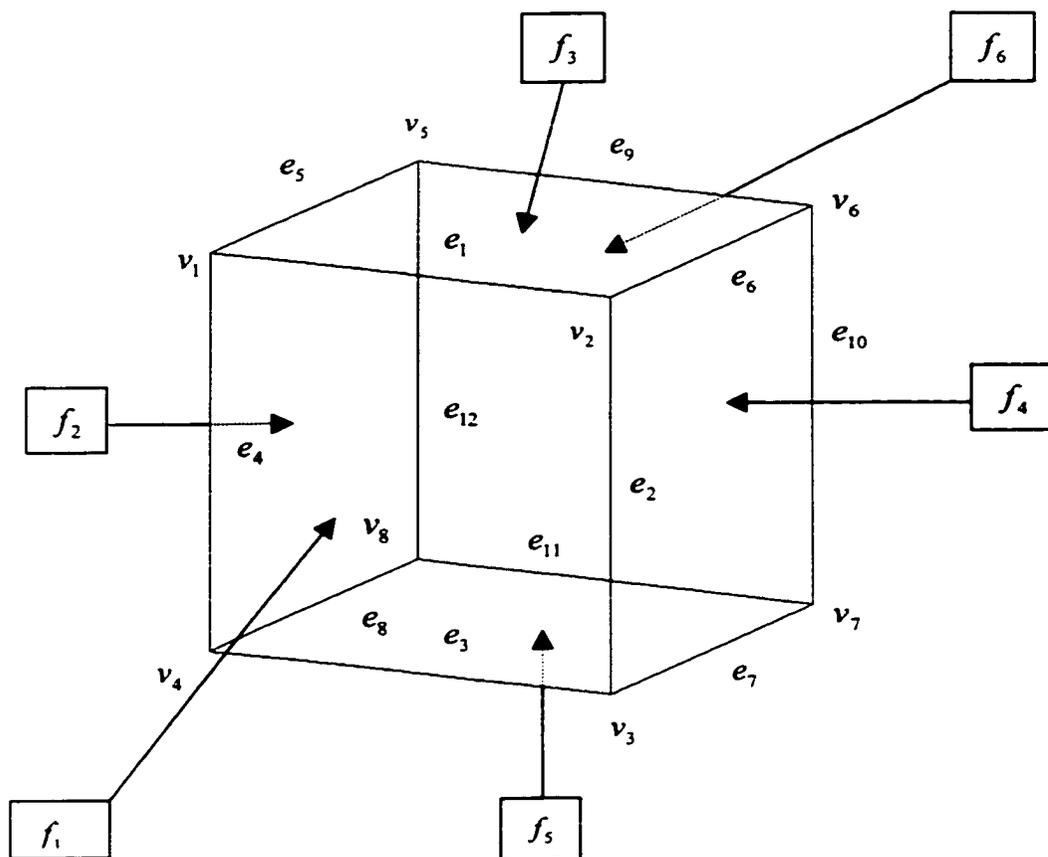


Figure 1.1: A labeled block solid

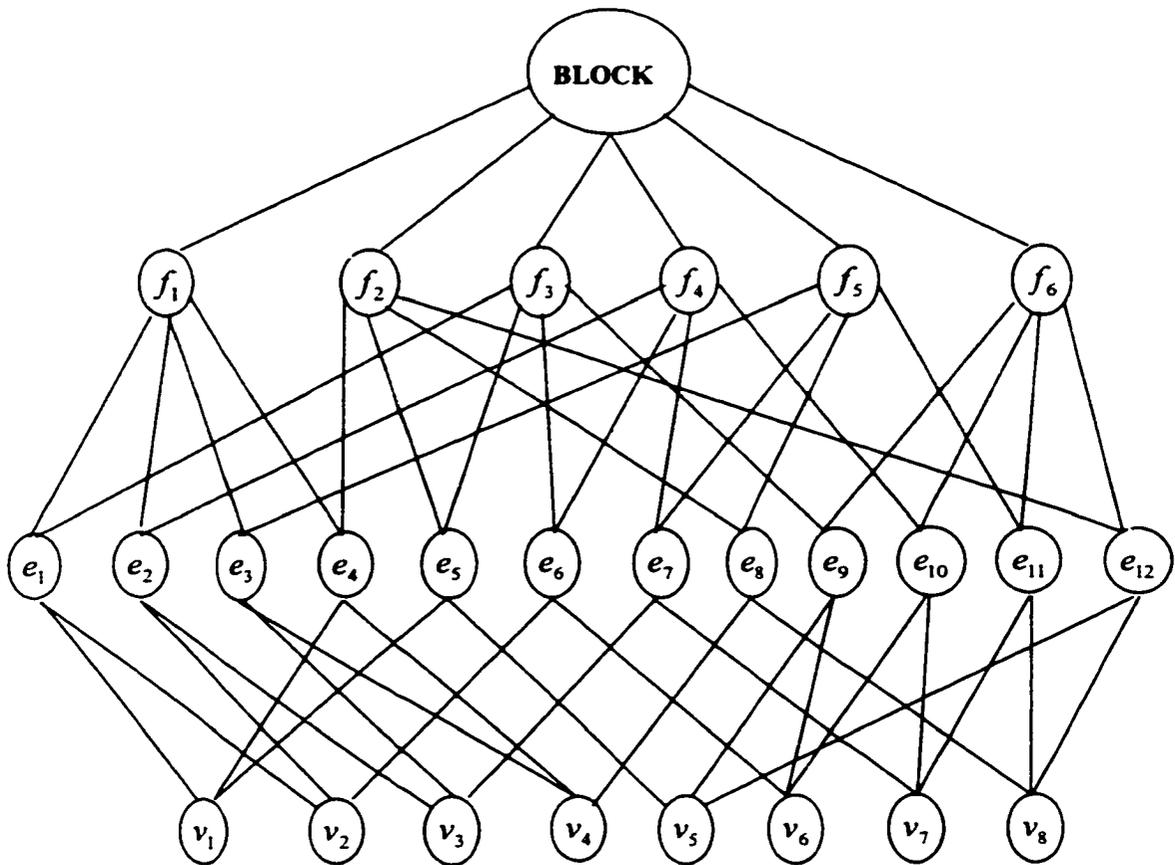


Figure 1.2: Winged-edge representation of the block in Figure 1.1

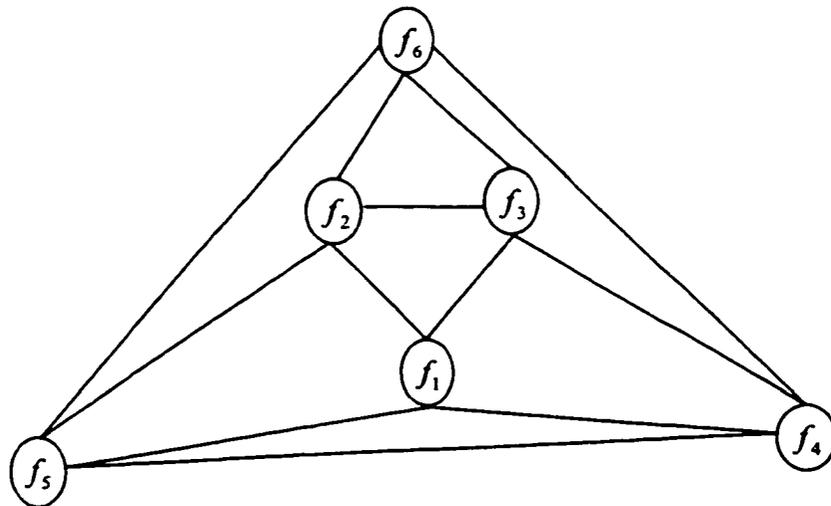


Figure 1.3: Face-edge graph of the block in Figure 1.1

Exhaustive enumeration uses the way of representing a binary picture in digital image processing. It decomposes an object into a set of small cubes of uniform size and orientation. The cubes are considered to be partially or completely inside the solid object. In this scheme, the object can be stored as a three dimensional matrix of binary data. With this binary matrix representation, many algorithms in digital image processing, such as edge detection, shape descriptor and connectivity can be applied on this binary representation to extract useful feature vectors for model description.

Space subdivision recursively decomposes the object into eight octants and represents it in the form of an 8-leaf tree (octree). This adaptive scheme is more efficient over uniform exhaustive enumeration. However, it is not so easy to apply digital image algorithms on this representation for its non-uniform scheme.

In the cell decomposition the basic units are in the form of polyhedrons with curved surfaces and can be more than one type. Various basic cells are glued together along their bounding surfaces.

Constructive model uses a set of unbounded half spaces to define an object. These half spaces are combined by regular Boolean set operations. Constructive Solid Geometry (CSG) is a variation of constructive models in which the set of unbounded half spaces is replaced by a set of primitive solids, such as blocks and cylinders, for the construction of an object.

CSG modelers store an object model in terms of the construction procedure. The data structure is a binary CSG tree. In the CSG tree, the leaves contain primitive solids and interior nodes are either Boolean operators or rigid transformations.

All of the above three representation schemes supply low-level product definitions in terms of low-level details, i.e. geometry and topology for B-Rep and primitives and operators

for CSG. They contain information that can totally determine locations of spatial points in relative to the objects and suitable for visualization and finite element applications. However these modeling data structures cannot be used to drive most computer aided engineering (CAE) systems in an automated manner since they do not supply a complete product definition. Different kinds of feature descriptors need to be explored for downstream applications, such as manufacturing evaluation (ME).

1.2.3 Feature Recognition

A significant amount of research has been performed in feature recognition from solid models since the late 1970s. The problems of shape feature recognition can be traced back to the seminal work of Kyprianou (1980) at the University of Cambridge. Kyprianou intended to develop an automated system capable of recognizing, classifying, and coding features of prismatic and rotational parts using feature grammars. Feature recognition techniques can be divided into two general classes: recognition from B-Rep models and recognition by volume decomposition algorithms.

For the B-Rep based approach, feature recognition can be interpreted as the grouping of the set of faces on the part boundary that has certain relationships between them. B-Rep based approach can be either rule-based or graph-based.

Hednerson (1984) developed a rule-based feature recognizer coded in Prolog, which describes features as heuristic rules. A solid model in boundary representation is converted into facts in Prolog. Then Prolog's pattern-matching mechanism recognizes features. Since it is difficult to write a rule for every specific feature, Henderson wrote rules for swept

features like generic holes, slots, and pockets. However, Henderson did not classify recognized holes into more specific types of holes.

Choi et al. (1984) described methods for recognizing generic holes, slots, and pockets in a 3D solid model. The definitions of regional shape patterns were embedded in a feature recognizing program written in Pascal. The program was implemented for recognizing generic holes.

Most rule-based approaches are time consuming, especially for 3D models with a relatively large number of topological entities. To address this problem, Gadh and Prinz (1992) developed a concept of cyclic loop. A cyclic loop, defined as a cyclic set of edges, is obtained by a Differential-Depth-Filter procedure. On the basis of cyclic loops, four feature classes are extracted: projection features, blind depression features, through depression features, and bridges.

Many researches have also resorted to graph-based approaches. In graph-based approaches, features are defined in terms of a graph of faces, edges, and vertices. The recognition technique searches for a pattern corresponding to the feature graph in the graph of the entire object.

De Floriani (1987) developed an edge-face graph for the extraction of protrusions and depressions on one or two object faces, and through holes or handles. The edge-face graph was decomposed into its bi-connected and tri-connected sub-graphs.

Joshi and Chang (1988) represented machined features with an Attributed Face Adjacency Graph (AFAG). AFAG was defined as a graph $G = [N, A, T]$ representing face adjacency and solid angles around edges. Where N is the set of nodes corresponding to the faces, A is the set of arcs corresponding to edges between two adjacent faces, and T is a set

of attributes assigned to the arcs depending on the edge convexity or concavity. This method also handles some cases of intersecting features by heuristic rules. However, the implementation of AFAG technique is limited to polyhedral features.

Sakurai and Gossard (1990) used sub-graph matching to interactively recognize user-definable shape features. The shape feature was defined as a set of contiguous faces possessing certain characteristics in topology, geometric types of faces, and solid angle around edges. Once the system recognizes a shape feature, it removes the geometry associated with the feature from the original solid model. The process repeats until no additional features are found.

With different choices of node and arc set from a complete adjacency graph, different kinds of adjacency sub-graphs can be constructed.

Chuang and Henderson (1990) developed the concept of the Vertex-Edge (V-E) graph for the recognition and classification of regional shape patterns. The nodes of the V-E graph are labeled by vertex types, which are determined by topology and geometric properties surrounding the vertices. In this research, vertices are classified differently in anticipation of their use in varying applications.

Laakko and Mäntylä (1993) developed an improved adjacency graph called surface-based attributed adjacency graph (SAAG). The SAAG interprets nodes as surfaces instead of faces in AFAG. A node data structure of the SAAG contains the following information: (1) a node identifier; (2) references to one or several adjacent solid model faces that belong to the geometrically same surface; (3) connectivity information; and (4) reference to a shape data structure. In the reported feature modeling system EXTDesign, four types of features were recognized incrementally.

Qamhiyah et al. (1996) introduced the concept of Loop-Adjacency Hyper Graph (LAHG) for the extraction of form features from solid models. The node in the graph represents a loop and the arc represents an edge. An arc is assigned a value +1, -1, or 0 depending on whether the edge is convex, concave, or belonging to an inner loop. The extraction and classification of form features is based on their effects on changing the basic shape.

A feature recognition system known as FeatureFinder was reported by Little et al. (1997) in Heriot-Watt University. FeatureFinder uses a machining direction based heuristic to decompose the face-edge graph into disconnected components. It can identify cylindrical depressions and protrusions.

Unlike B-Rep based feature recognition, volume-based approaches extract features by using the volumetric properties of geometric models.

Lee et al. (1987) used tree manipulation techniques to rearrange an object's CSG representation into CSG patterns corresponding to solid features. This approach is based on the observation that most features can be associated with a very limited number of axes of symmetry. Features are defined as CSG combinations of primitives whose axes of symmetry satisfy certain geometric relationships.

Woo (1982) developed a volume-based method called alternating sum of volumes (ASV) decomposition. The volume of the object is decomposed by subtracting it from its convex hull. Kim and Wilde (1992) incorporated ASV decomposition with remedial partitioning to obtain a convergent convex decomposition technique called Alternating Sum of Volumes with Partitioning (ASVP). The geometric domain of ASVP decomposition was extended to include cylindrical features and constant-radius blending features [Wang, 1997].

Feature recognition approaches can successfully identify objects with isolated features and features with planar surfaces or cylindrical surfaces. However, there are still many challenging problems, which have been addressed on a special panel session for feature recognition at the 17th ASME international computers in engineering conference. Han et al. (1997) pointed out that future feature recognition research should focused on the following four issues:

- (1) Determining form features with free-form surfaces. One way to recognize such features is linearizing the non-linear surfaces. However, linearization increases the number of topological entities significantly.

Sonithi and Gadh (1997) developed a Curvature Region (CR) approach for the extraction of form features with free-form surfaces. A B-Rep model was converted to a CR representation based on curvature properties of B-Rep entities (vertices, edges, and faces). Two primitive shape classes (protrusions and depressions) are obtained in terms of CR representation. Features are recognized by using these two primitives. In Sonithi and Gadh's work, a free-form face tends to be broken into different small regions.

- (2) Recognizing intersecting features

Regli (1996) has concentrated on this topic and tried to refine what is feature interaction. Only based on the spatial relationships, feature interactions can be classified into the following four types: (1) Interference Interaction—features may have their surfaces overlapped, or there is some volume shared by each other; (2) Adjacency Interactions—the feature volumes coincide over part of their boundaries; (3) Remote Interactions—the features are totally disjointed, not adjacent or overlapping, but there is some functional relational relationship or they are both simultaneously significant to

some downstream process of design; (4) Life Cycle Interactions—feature interactions among feature models in multi-application areas through the life cycle.

As an example, Han et al. (1997) reported a hint-based algorithm to deal with complex feature intersections. Han et al. believed that a feature should leave a trace in the part boundary even when features intersect. However, this method works mainly on the machining domain, and in some cases, a significant number of hints do not necessarily lead to a valid feature.

- (3) Providing desirable feature interpretation for a specific application domain
- (4) Incorporating manufacturing knowledge, such as manufacturing resources and tooling into features.

1.2.4 Shape Analysis and Description

Shape analysis has been an active research issue in digital image encoding and visual pattern recognition for the past three decades. Shape-based description is one of the two factors conditioning the performance of a computer vision recognition system [Fan 1989]. For digital images, differential-geometric method and transform-based local extraction method are two of the most popular mathematical methods to generate the shape description.

Differential-geometric method deals with the definition and estimation of characteristic features in terms of differential geometry. It provides transformation-invariant differential shape descriptors for surface characterizations, such as Gaussian curvature that is invariant under rigid motions.

Persoon and Fu (1977) used the shape boundary curves for object description and classification. In their work, Fourier descriptors that are invariant to rotation, translation, and

scaling were used to represent object shapes. These Fourier descriptors were tested in machined part recognition. In the test, a part's silhouette was extracted by thresholds and converted by Fourier descriptors. The proposed recognition algorithm was then applied on the output binary matrix. Parts with minor variations in shapes are classified into the same class and the transformation of the part does not influence the performance of the recognition algorithm.

Eichmann and Jankowski (1985) extended the work of Fourier descriptors to represent and measure surfaces in three-dimensional closed volume. A new shape descriptor was generated based on the solid angle. In Eichmann and Jankowski's work, the term "solid angle" was defined as the projection area of a closed volume's surfaces upon the surface of a unit sphere.

Van Otterloo (1991) used parametric contours to represent segmented images. Contours were then analyzed and classified by similarity and symmetry measurement methods. Such similarity measures were based on the coefficients of Fourier or Walsh transforms.

Fan et al. (1987, 1989) introduced a method to describe surfaces of three-dimensional objects by surface patches caused by surface discontinuities. Using the zero-crossings and extrema of curvatures, the surface was segmented into simple surface patches. And then these patches and their boundaries were used to describe the three-dimensional surfaces. The surface patches were generated by the following three different points and lines that correspond to different physical properties:

- (1) Jump boundary, caused by surface discontinuity, creates zero-crossing of the curvature in a direction normal to that of the boundary;

(2) Fold, caused by surface orientation discontinuity, creates a local extrema of the curvature;

(3) Ridge line, creates a local extremum of curvature.

With a priori knowledge of the feature being searched for, template methods are also available for feature detection and representation. Yuille, Cohen, and Hallinan (1989) developed deformable templates to describe features of faces. The starting changeable templates were set up as expected shapes. The templates then interacted with face image dynamically to get a best fit by minimizing the energy function. The final values of template parameters were used as features' descriptors.

Wavelets can represent an object by its local approximation and detail in multi-scale scheme. This multi-resolution representation is very effective for analyzing and coding the information content in images [Mallat 1991, Swanson 1996, Prasad et al. 1997].

Tieng and Boles (1997) presented a wavelet-based algorithm for recognizing a 2D-closed boundary object. The object boundary was traced and represented by a 1D signal. The first derivative of a cubic spline was proposed as the basic wavelet. The zero crossings of the wavelet transform were used to extract the signal's local variation patterns.

Tieng and Boles (1997) further extended this wavelet-based affine invariant method to represent and recognize planar objects in a three-dimensional space. Three wavelet families (Daubechies, Lemaire, and B-Spline) were used to extract multi-resolution affine invariant features from the parameterized contour representation. A dissimilarity function was developed for matching objects to models in the database.

Alferez et al. (1999) presented shape descriptors invariant to both general affine transformation and the change of illumination. For transformation invariant, the contour of

object images were properly extracted and used for representation. For illumination invariant, shape internal regions were linearized by a characteristic curve and color signatures along characteristic curve were computed for the representation. The shape invariant could be measured at different resolution scales and were applicable to many basis functions, such as wavelet, short time Fourier analysis, and spline bases.

As an application in medical imaging and tomography, Devore et al. (1996) developed new wavelet-based techniques for the automatic detection of certain early signs of breast cancer. The authors focused on one of the most important signs of breast cancer—cluster of micro-calcifications that appear as small bright spots in the images. Wavelet transforms were applied on the mammographic images with size 512×512 . Only those terms at three highest levels were kept before the image was reconstructed. The shapes of micro-calcifications were detected on the reconstructed image.

Wavelets have also been used in 3D-object recognition from 2D images. Through Gabor wavelet decomposition of a series of 2D images of an object, Wu and Bhanu (1997) built a 3D model of the object. The magnitude, phase, and frequency parameters of the Gabor wavelet representations were then used to localize objects in image.

Wavelet transform was also used for analyzing texture in digital images. GroB et al. (1994) took the localization properties of wavelet transform and developed image feature vectors by a set of wavelet coefficients in the neighborhood of a pixel.

Another biggest application of shape analysis is in the evaluation of computer-aided geometric design in a CAD/CAM System. Beck et al. (1986) explored several tools for investigating the intrinsic shape properties of curved free-form surfaces. The contour maps and high-resolution shaded images were used to visualize and detect surface shapes. The

curvature analysis provides guidance for a NC programmer in choosing cutter dimensions for gouge-free milling of free form surfaces.

Rando and Roulier (1991) developed a shape fairness algorithm for automatically evaluating and adjusting designed parametric surfaces by using a collection of fairness metrics. The shape fairness metrics were geometric measures of surface fairness depending only on the shape of the surface. The metrics were derived from the area of the surfaces that were constructed from the invariants of the designed surfaces. In the proposed algorithm, the fairness metrics was used as an objective function for some standard minimization.

Brlyayev and Ohtake (1999) extended 2D image edge detection tools to extract ridges on surfaces approximated by a triangular mesh. Ridges are defined by the extrema of surface curvatures and used for shape description and segmentation. Ridges are the locus of points where the maximum principal curvature attained a positive maximum along its curvature line.

1.3 Research Objective

The review in Section 1.2 can be concluded with the following two observations on the geometric modeling:

- (1) Geometric models represent an object in terms of low-level entities and have proved to be useful for geometry detailing, visualization, and FEA analysis. However, geometric models are no longer optimal representations for incorporating Design for Manufacturability (DFM) evaluation tools into the early design process;

(2) Recognizing domain-independent form features in geometric models is a promising solution for CAD/CAM integration, and has been an active research area for decades. However, feature recognition techniques are currently facing challenging problems, such as intersecting features and features with free-form surfaces.

The objective of this dissertation is the development of a general mathematical framework for form features identification and coding. In general, there are two stages for automatic feature recognition and classification: (1) describing form features in terms of pure geometry and topology in a CAD model, (2) establishing the correspondence between form feature descriptions and manufacturing features by incorporating manufacture resource information.

The research described in this dissertation focuses on the first stage by generating surface-based representations that are used as an input to a form feature extraction procedure. A surface is chosen as a representation primitive as it can capture both the designer's intentions and manufacturing resource information. The designer specifies the shape of a product in terms of its surfaces and dimensions the product by relating its surfaces to one another. The manufacturer considers a manufacturing feature as a group of surfaces that can be produced simultaneously.

Each single face instead of multiple faces like most previous B-Rep based feature is handled for its process requirements at a time. A set of connected faces, which share a common manufacturing tools and can be simultaneously manufactured, are then grouped together as a manufacturing feature.

To expand the domain of feature extraction to include products with non-planar surfaces, a solid model is digitized by uniform cells and represented in a binary three-dimensional

matrix. In this binary representation, form features can then be interpreted as singularities in a product's "shape signal." A singularity is an abrupt change in a shape signal.

Three-dimensional wavelet transform (3D WT) is used in this research for the shape analysis and coding of the digitized representation. 3D WT can decompose a three-dimensional signal into a set of orthogonal, spatially oriented representations. When 3D WT is applied to the object's binary matrix, it can detect the sharp contrast caused by a feature's surfaces and assign different detail coefficients according to the surfaces' curvature patterns.

1.4 Overview of Chapters

Chapter 1 is an introduction that gives a general description of form feature, geometric models, feature recognition and shape analysis. The theoretical aspects of wavelet transforms used for shape feature analysis and coding are presented in Chapter 2. Different orthogonal wavelet basis, pyramid decomposition algorithm and three-dimensional extension of wavelet basis are discussed.

Chapter 3 presents a novel wavelet-based surface analysis and classification algorithm. Two wavelet bases, Harr and Daubechies with different vanishing moments, have been implemented. Examples are presented to illustrate the proposed procedure.

Chapter 4 describes a face-based coding procedure that is applicable to the object and its extracted features. The overall conclusions of the work and future research recommendations are presented in Chapter 5.

CHAPTER 2

WAVELET DECOMPOSITION

2.1 Wavelet in Comparison with Windowed Fourier Transforms

To reveal and characterize signal structures of different patterns and sizes, it is necessary to use time-frequency analysis. Windowed Fourier transforms and wavelet transforms are two important time-frequency methods. Windowed Fourier transforms use a single pure harmonic window for all frequencies. The resolution of the windowed Fourier Transform decomposition is the same at all locations across the time-frequency plane. Figure 2.1 illustrates a windowed Fourier transform, where the window is simply a square wave. The square wave window localizes the sine and cosine base functions to particular neighborhoods in time. In order to achieve a high degree of localization in time and frequency, a function with sufficiently narrow time and frequency windows should be chosen. However, the measurement of time and frequency components of the window function is lower bounded by the Heisenberg's Uncertainty Principle, which implies that the measurement resolution in both the time and frequency domains cannot be arbitrarily small at the same time. It is well known that the Gaussian function, which is used as the window for Gabor transform, is the tightest time-frequency window of all windowed Fourier transforms.

The wavelet transform is a multi-resolution analysis (MRA) method that filters a data sequence into different frequency intervals and analyzes each frequency piece-by-piece with a resolution matched to its scale. The advantage of the wavelet transforms over the

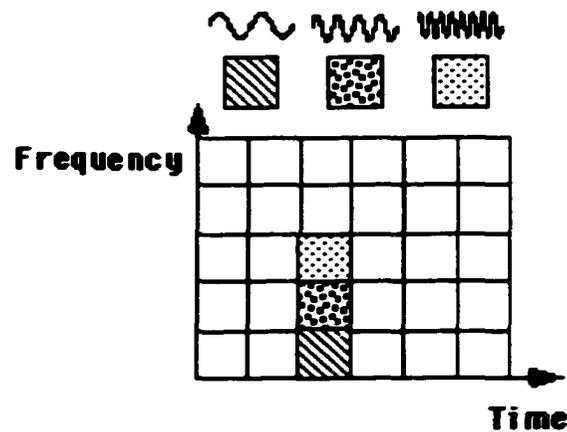


Figure 2.1: Windowed Fourier transforms in time frequency plane

windowed Fourier transform is its time-frequency or spatial-frequency localization properties. The wavelet transforms are designed based on a constant Q concept [Rioul and Vetterli, 1991], which states that the ratio of the central frequency to the width of time-frequency window is a constant.

Unlike the windowed Fourier transform which has a constant resolution at all times and frequencies, the WT has a good time and poor frequency resolution at high frequencies, and good frequency and poor time resolution at low frequencies. In order to get a detailed view of the signal, wavelet transforms use some very short-time and high-frequency basis functions. On the other hand, in order to obtain the global view of the signal, wavelet transform would have some long-time and low-frequency basis functions. Figure 2.2 shows the coverage in the time-frequency plane with the Daubechies wavelet, which will be discussed later in this chapter.

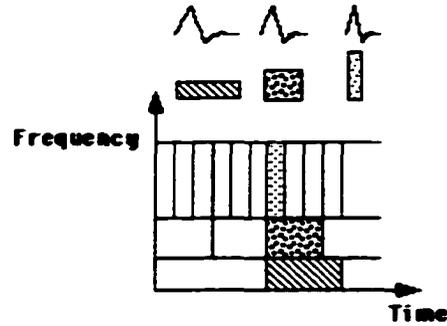


Figure 2.2: Wavelet transforms in time frequency plane

A family of wavelet transform base functions is generated from a basic wavelet function $\varphi(t)$ called mother wavelet by scaling $\varphi(t)$ by s and translating it by u :

$$\varphi_{u,s}(t) = \frac{1}{\sqrt{s}} \varphi\left(\frac{t-u}{s}\right) \quad (2.1)$$

In order for a function f to be recovered from its wavelet transform, the mother wavelet $\varphi(t)$ should satisfy the following admissibility condition:

$$\int_{-\infty}^{\infty} |\hat{\varphi}(w)|^2 |w|^{-1} dw < \infty \quad (2.2)$$

Where $\hat{\varphi}(w)$ denotes the Fourier transform of $\varphi(t)$.

The operations of the wavelet transform are similar to those of matching templates. The wavelet decomposition of a signal can be viewed as a series of templates operating on the signal, with the size of templates varying for each level. If local details of the signal waveform match the working template, the output coefficients are maximized, and otherwise the coefficients are almost zeros. Short-time filters give fine resolutions and long-time filters give coarse resolutions. The combination of the results for the entire time-frequency plane provides the final output of wavelet transform. When applied to the digital geometry of a

product, the wavelet transform acts as a multi-resolution filter that is capable of zooming in onto the well-localized geometrical detail and then zooming out onto the global pattern.

2.2 Fundamentals of the Wavelet Transforms

Basically, there are two types of wavelet transform: (1) Continuous wavelet transform, and (2) Discrete wavelet transform. The discrete wavelet transform can be further classified as: (1) Redundant discrete systems (frames), and (2) Orthonormal (and other) bases of wavelets.

2.2.1 Continuous Wavelet Transform

In order to achieve varying time and frequency windows, the continuous wavelet transform was developed as an alternative approach to the windowed Fourier transform. The wavelet analysis is done in a similar way to the windowed Fourier transform by projecting the signal onto base functions. Similar to the window function in the windowed Fourier transform, the continuous wavelet transform is computed locally for each segment of the time-domain signal. However, the width of time-frequency window is changed as the continuous wavelet transform is computed for every spectral component.

The continuous wavelet transform of a function $x(t)$ is defined as follows:

$$(CWT_{\phi}x)(u,s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} x(t) \phi^* \left(\frac{t-u}{s} \right) dt \quad (2.3)$$

Where “*” indicates the complex conjugate.

The computation of the continuous wavelet transform starts with a specific scale s . For the scale s , the wavelet $\varphi(t)$ is placed at the beginning of the signal and the above equation (2.3) is computed. Then the wavelet is shifted towards the right by a sufficiently small step u and the integration is re-computed. The procedure is repeated until the end of the signal. Finally the computation for the whole row of points on the time-frequency plane for the scale s is completed.

2.2.2 Discrete Wavelet Transforms

Continuous wavelet transforms can be sampled into wavelet series for the numerical solutions. However continuous wavelet transforms provide highly redundant information and require a significant amount of computation time and computer resources. The discrete wavelet transforms, on the other hand, provide sufficient information both for analysis and synthesis of the original signal, with a significant reduction in the computation time. The pyramid algorithm and multi-resolution signal analysis developed in the early eighties built the foundation of the discrete wavelet transform [Burt and Adelson, 1983].

2.3 Redundant Discrete Wavelet Transform Frames

For the discrete wavelet transform, $s = s_0^m$ and $u = n u_0 s_0^m$, where $m, n = 1, 2, 3, \dots$, $s_0 \neq 1$ and $u_0 > 0$. This corresponds to

$$\varphi_{m,n}(t) = \frac{1}{\sqrt{s_0^m}} \varphi\left(\frac{t - n u_0 s_0^m}{s_0^m}\right) = s_0^{-m/2} \varphi(s_0^{-m} t - n u_0) \quad (2.4)$$

In order for the numerically stable reconstruction of the function f from its decomposition sequence $\langle f, \varphi_{m,n}(t) \rangle$, $\varphi_{m,n}(t)$ should be able to constitute a frame. A frame is defined as (Daubechies, 1992):

A family of functions $(\varphi_i)_{i \in J}$ in the Hilbert space is called a frame if there exist $A > 0$, $B < \infty$ so that, for all f in the Hilbert space,

$$A\|f\|^2 \leq \sum_{i \in J} |\langle f, \varphi_i \rangle|^2 \leq B\|f\|^2 \quad (2.5)$$

Where A and B are the frame bounds.

2.4 Orthonormal Wavelet Bases

The search for orthonormal wavelets begins with multi-resolution signal approximations. Mallat (1989) defined the discrete wavelet theory as a multi-resolution approximation of signals at various scales with projections on a sequence of successive orthogonal bases $\{v_j\}_{j \in Z}$. The orthonormal approximation spaces $\{v_j\}$ should satisfy:

$$\dots v_2 \subset v_1 \subset v_0 \subset v_{-1} \subset \dots \quad (2.6)$$

$$\overline{\bigcup_{j \in Z} v_j} = L^2(\mathfrak{R}) \quad (2.7)$$

$$\bigcap_{j \in Z} v_j = \{0\} \quad (2.8)$$

$$f \in v_j \Leftrightarrow f(2^j \cdot) \in v_0 \quad (2.9)$$

$$f \in v_0 \Rightarrow f(-n) \in v_0 \text{ for all } n \in Z \quad (2.10)$$

Where Z denotes integer, and $L^2(\mathfrak{R})$ represents finite energy functions $\int |f(t)|^2 dt < +\infty$.

In addition, if there exists $\varphi \in v_0$, $\{\varphi_{0,n}; n \in Z\}$ should be an orthonormal basis in v_0 .

The approximation is computed by projecting the signal into a family of orthogonal functions, which are the translation and dilation of a basic scaling function ϕ . The difference of information between the approximation of the signal at the two adjacent scales can be extracted by projecting this signal on a wavelet orthogonal basis.

The approximation of f at base v_k ($v_k = span\{\phi_j^k(x)\}, j = -\infty \rightarrow \infty$) is:

$$S_k f(x) = \sum_j \langle f, \phi_j^k(x) \rangle \phi_j^k(x) \quad (2.11)$$

Where $\phi_j^k(x) = 2^{-k/2} \phi(2^{-k}x - j)$ and \langle , \rangle denotes the inner product.

The complement of v_{i+1} in v_i can be obtained by decomposing the signal into a base w_{i+1} ($w_{i+1} = span\{\varphi_j^{i+1}(x)\}, j = -\infty \rightarrow \infty$), which are the translation and dilation of the mother wavelet φ . As a result,

$$D_k f(x) = \sum_j \langle f, \varphi_j^k(x) \rangle \varphi_j^k(x) \quad (2.12)$$

Where $\varphi_j^k(x) = 2^{-k/2} \varphi(2^{-k}x - j)$

Vanishing moments are used to describe the regularity of a scaling function ϕ and wavelet φ . ϕ and φ become more regular as the number of vanishing moments increases. Large vanishing moments will produce small evenly distributed detail coefficients and good approximation properties.

The k^{th} moment of φ is defined as

$$M_k = \int x^k \varphi(x) dx \quad (2.13)$$

φ has N vanishing moments if

$$\int x^k \varphi(x) dx = 0 \quad k = 0, \dots, N-1 \quad (2.14)$$

At least one vanishing moment is necessary for ϕ and φ to exist. That also means a wavelet is a function with a zero average, $\int \varphi(x) dx = 0$.

Different classes of wavelet bases have been proposed and constructed in previous studies (Daubechies 1988, Chui 1992, Daubechies 1993). The primary goal of designing a wavelet base is to optimize it to produce a maximum number of wavelet coefficients that are close to zero. Mallat (1998) considered as key factors the number of vanishing moments of wavelet, the size of its support, and the regularity of input signal. Orthonormal wavelet based can be constructed with either infinitely or compactly supported function. In the following sections, the infinitely supported Battle-Lemarie wavelets, the Harr wavelets and the Daubechies wavelets are presented.

2.4.1 Battle-Lemarie Wavelets

Battle and Lemarie introduced polynomial splines for constructing scaling functions. The multi-resolution approximations are built from polynomial splines of order $2p + 2$ [Mallat, 1989]. Lemarie has shown that the scaling function associated with such a mutli-resolution approximation can be written as:

$$\phi(\omega) = \frac{1}{\omega^n \sqrt{\sum_{2^n}(\omega)}} \quad (2.15)$$

Where $n = 2p + 2$ and $\sum_n(\omega) = \sum_{k=-\infty}^{+\infty} \frac{1}{(\omega + 2k\pi)^n}$

Quadrature mirror filter $H(\omega)$ can be obtained from the following relation:

$$\hat{\phi}(2\omega) = H(\omega)\hat{\phi}(\omega) \quad (2.16)$$

The Following is a Battle-Lemarie example with $p = 1$ and thus $n = 4$. It corresponds to a multi-resolution approximation built from cubic splines. The scaling function for this example is:

$$\phi(w) = \frac{1}{w^4 \sqrt{\sum_8(w)}} \quad (2.17)$$

where

$$\sum_8(w) = \frac{N_1(w) + N_2(w)}{105(\sin \frac{w}{2})^8} \quad (2.18)$$

with

$$N_1(w) = 5 + 30(\cos \frac{w}{2})^2 + 30(\sin \frac{w}{2})^2 (\cos \frac{w}{2})^2 \quad (2.19)$$

and

$$N_2(w) = 2(\sin \frac{w}{2})^4 (\cos \frac{w}{2})^2 + 70(\cos \frac{w}{2})^4 + \frac{2}{3}(\sin \frac{w}{2})^6 \quad (2.20)$$

Battle-lemarie wavelets have the properties of symmetry, smoothness and exponential decay. However, Battle-Lemarie wavelet bases are infinitely supported. Therefore, their filters need to be truncated for the computation.

2.4.2 Harr Wavelet Bases

Harr is the only symmetric wavelet with a compact support. Its scaling function and mother wavelet are defined by:

$$\phi(x) = \begin{cases} 1, & \text{for } 0 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (2.21)$$

and

$$\varphi(x) = \begin{cases} 1, & \text{for } 0 \leq x \leq \frac{1}{2} \\ -1, & \text{for } \frac{1}{2} \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (2.22)$$

Harr wavelet is real and anti-symmetric about $x=1/2$. In first level decomposition, averaging every two neighbors does the approximation, and details are obtained by differentiating every two adjacent points. Harr wavelet has optimal localization properties in the spatial domain, which makes it very useful for capturing small details in 3D-shape analysis. However, Harr wavelet is not continuous and has bad localization in the frequency domain.

2.4.3 Daubechies Compactly Supported Wavelet Bases

Daubechies (1988) set out to find compactly supported wavelets with the maximum number of vanishing moments for a given support width. Compactly supported wavelets sacrifice symmetry and continuity in exchange for spatial localization. Compacted supported wavelets can be implemented by conjugate mirror filters: the low-pass filter $h[n]$ and the high-pass filter $g[n]$. The low-pass filter is:

$$h[n] = \frac{1}{\sqrt{2}} \langle \phi\left(\frac{t}{2}\right), \phi(t-n) \rangle \quad (2.23)$$

and the relation between $h[n]$ and $g[n]$ is:

$$g[n] = (-1)^n h[2N - n - 1] \quad (2.24)$$

In order to find the most compact wavelets for a number of given vanishing moments, Daubechies [1988] considered a polynomial of minimum degree m such that $\hat{h}(w)$ satisfies

$$|\hat{h}(w)|^2 + |\hat{h}(w + \pi)|^2 = 2 \quad (2.24)$$

Here
$$\hat{h}(w) = \sum_{n=0}^{N-1} h[n] e^{-inw} \quad (2.25)$$

Daubechies proved that the minimum degree is $m = N - 1$. The size of the resulting filter h is $2N$. When $N = 1$ we get the Harr wavelet.

For the given length of the mother wavelet, Daubechies filters maximize the smoothness of the scaling function by maximizing the rate of decay of its Fourier transform. There could exist compactly supported mother wavelet ϕ and scaling function φ with more regularity or smoothness, but with larger support lengths for ϕ, φ .

The conjugate mirror filter coefficients for Daubechies wavelets ranging from one to five vanishing moments are listed below in Table 2.1.

In general, there are no close analytic formula for compact supported ϕ, φ , except when $N=1$ (Daubechies 1992). Given the conjugate mirror filter coefficients, the scaling function for the Daubechies wavelets can be approximated with arbitrarily high precision by the cascading functions:

$$\phi(t) = \sqrt{2} \sum_{k=-\infty}^{+\infty} h_k \phi(2t - k) \quad (2.26)$$

The Daubechies wavelets with 2 vanishing moments, denoted by DAUB4, are used for an example. The recursion equations for the scaling function and the wavelet function are:

$$\phi(t) = h_0\phi(2t) + h_1\phi(2t - 1) + h_2\phi(2t - 2) + h_3\phi(2t - 3)$$

$$\begin{aligned}\varphi(t) &= g_0\varphi(2t) + g_1\varphi(2t - 1) + g_2\varphi(2t - 2) + g_3\varphi(2t - 3) \\ &= h_3\varphi(2t) - h_2\varphi(2t - 1) + h_1\varphi(2t - 2) - h_0\varphi(2t - 3)\end{aligned}$$

Figures 2.3 and 2.4, Figures 2.5 and 2.6 show the plots of the scaling function ϕ and wavelet function φ for DAUB4 and DAUB8 respectively.

2.5 The Pyramid Algorithm for DWT Decomposition

The discrete wavelet transform is a linear operation that decomposes a signal into components at different scales. The DWT is implemented by the convolution of the signal with dilated filters. The signal is passed through the high pass filter $g[n]$ to analyze the high frequencies, and it is passed through the low pass filters $h[n]$ to analyze the low frequencies.

The resolution of the signal is a measure of different detail information in the signal.

The resolution is changed by up sampling \uparrow and down sampling \downarrow filter operations. Up sampling a data sequence corresponds to increasing its sampling rate by adding new samples to the sequence. For example, up sampling by two means to insert a new sample, usually a zero or an interpolated value, between every two samples of the sequence. Down sampling a sequence corresponds to discarding some of its samples. For example, down sampling by two refers to dropping every other sample of the sequence.

Table 2.1: The low filter coefficients $h[n]$ for compact supported Daubechies wavelets

Vanishing Moments (N)	n	Filter coefficients
1	0	0.707106781186
	1	0.707106781186
2	0	0.482962913145
	1	0.836516303739
	2	0.22413868042
	3	-0.129409522551
3	0	0.332670552950
	1	0.806891509311
	2	0.459877502118
	3	-0.135011020010
	4	-0.085441273882
	5	0.035226291886
4	0	0.230377813309
	1	0.714846570553
	2	0.630880767940
	3	-0.027983769417
	4	-0.187034811719
	5	0.030841381836
	6	0.032883011667
	7	-0.010597401785
5	0	0.160102397974
	1	0.603829269797
	2	0.724308528438
	3	0.138428145901
	4	-0.242294887066
	5	-0.032244869585
	6	0.077571493840
	7	-0.006241490213
	8	-0.012580751999
	9	0.003335725286

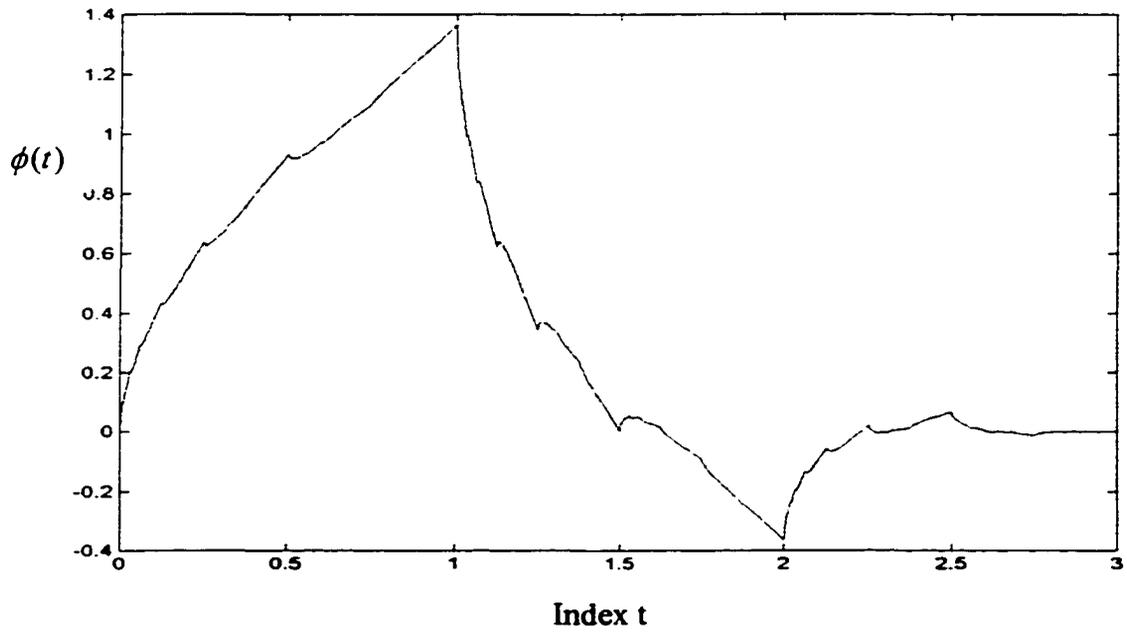


Figure 2.3: The scaling function $\phi(t)$ of DAUB4 wavelet transform

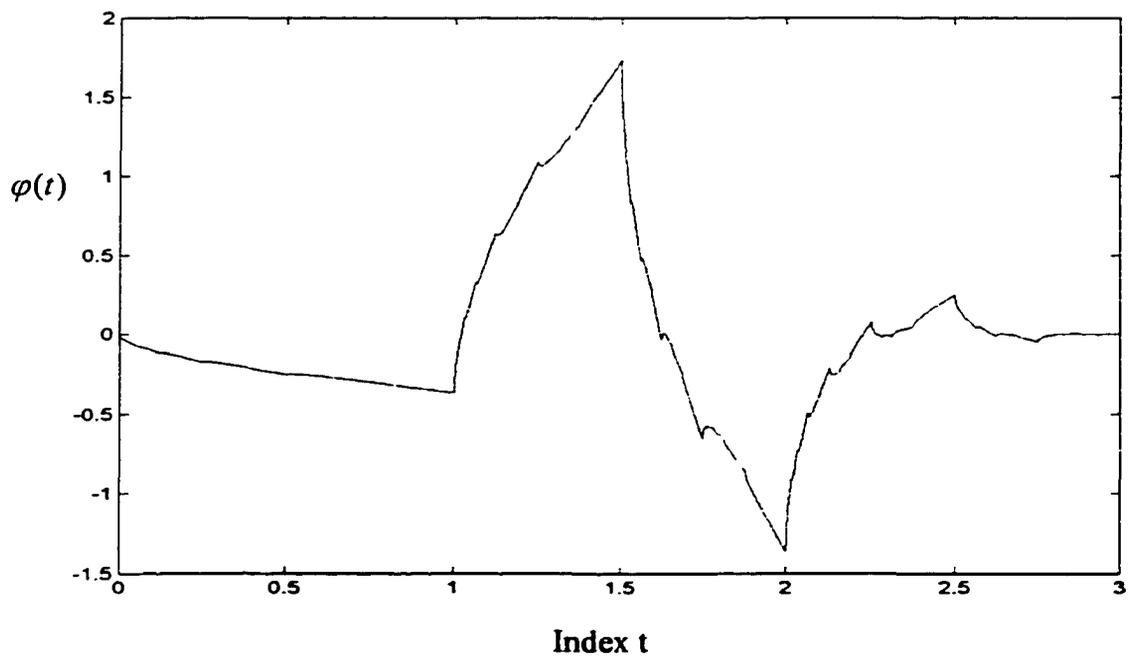


Figure 2.4: The mother wavelet $\psi(t)$ of DAUB4 wavelet transform

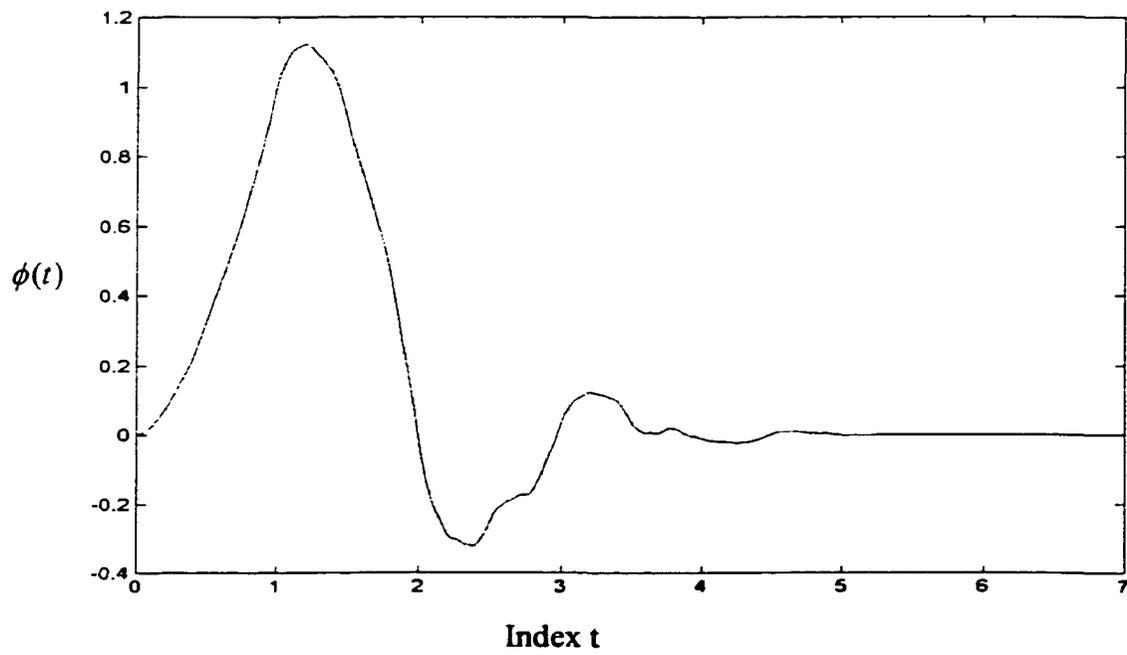


Figure 2.5: The scaling function $\phi(t)$ of DAUB8 wavelet transform

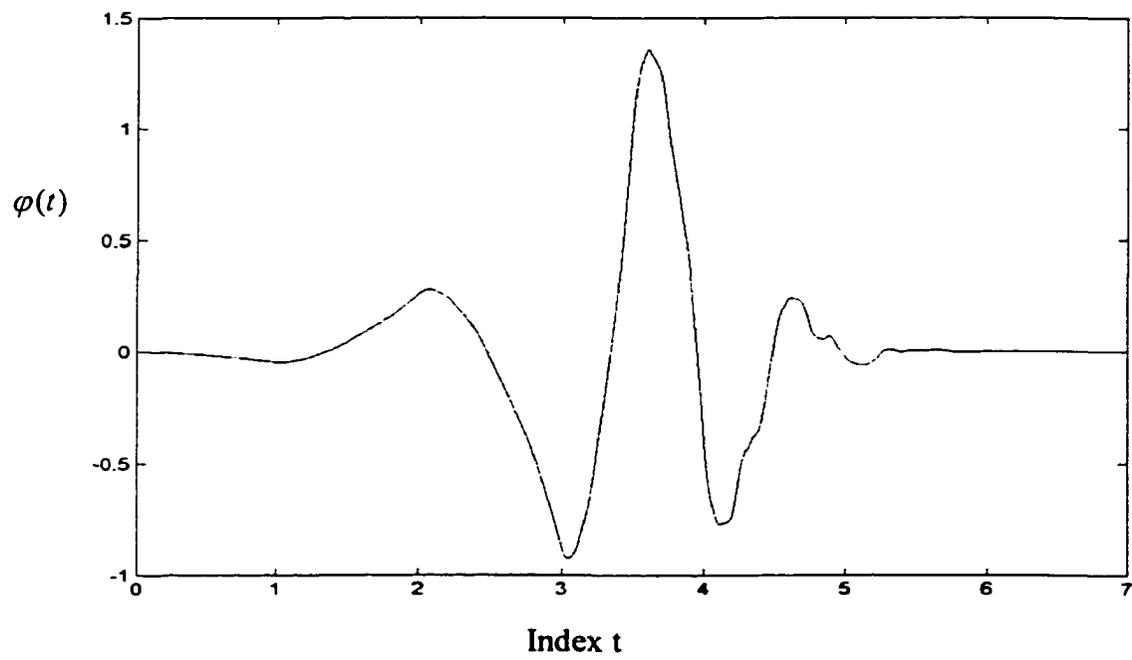


Figure 2.6: The mother wavelet $\phi(t)$ of DAUB8 wavelet transform

$$\begin{bmatrix} h_0 & h_1 & h_2 & h_3 & 0 & 0 & 0 & 0 \\ h_3 & -h_2 & h_1 & -h_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & 0 & 0 \\ 0 & 0 & h_3 & -h_2 & h_1 & -h_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \\ 0 & 0 & 0 & 0 & h_3 & -h_2 & h_1 & -h_0 \\ h_2 & h_3 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ h_1 & -h_0 & 0 & 0 & 0 & 0 & h_3 & -h_2 \end{bmatrix} \begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \end{bmatrix} = \begin{bmatrix} s_0 \\ d_0 \\ s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \end{bmatrix} \xrightarrow{\text{re-order}} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

Figure 2.7: The first level pyramid decomposition with DAUB4 for a vector of size 8

2.6 Three Dimensional Extension of Wavelet Bases

The research in this dissertation represents a product model by three-dimensional matrix. In order to analyze a product model in terms of such a matrix, it is necessary to extend one-dimensional bases to bases in $L^2(R^3)$.

Multi-dimensional separable wavelets can be constructed by tensor products of ϕ and φ generated by three one-dimensional bases. For n-dimensional wavelet orthonormal bases, there are $2^n - 1$ wavelets. For three-dimensional wavelets, we define 3D space V_i and space W_i that is the orthogonal compliment of V_i . We have

$$\begin{aligned}
V_j &= v_j^x \otimes v_j^y \otimes v_j^z = (v_{j+1}^x \oplus w_{j+1}^x) \otimes (v_{j+1}^y \oplus w_{j+1}^y) \otimes (v_{j+1}^z \oplus w_{j+1}^z) \\
&= \underbrace{v_{j+1}^x \otimes v_{j+1}^y \otimes v_{j+1}^z}_{\oplus} \underbrace{v_{j+1}^x \otimes v_{j+1}^y \otimes w_{j+1}^z}_{\oplus} \underbrace{v_{j+1}^x \otimes w_{j+1}^y \otimes v_{j+1}^z}_{\oplus} \underbrace{w_{j+1}^x \otimes v_{j+1}^y \otimes v_{j+1}^z}_{\oplus} \\
&\quad \oplus \underbrace{v_{j+1}^x \otimes w_{j+1}^y \otimes w_{j+1}^z}_{\oplus} \underbrace{w_{j+1}^x \otimes v_{j+1}^y \otimes w_{j+1}^z}_{\oplus} \underbrace{w_{j+1}^x \otimes w_{j+1}^y \otimes v_{j+1}^z}_{\oplus} \underbrace{w_{j+1}^x \otimes w_{j+1}^y \otimes w_{j+1}^z}_{\oplus} \\
&= V_{j+1} \oplus W_{j+1}
\end{aligned} \tag{2.28}$$

Here, x, y, and z stand for three coordinate directions.

Equation (2.28) shows that at each decomposition level, the space is broken up into one approximation subspace and seven orthonormal complementary subspaces in principal directions. Figure 2.8 visualized the basic scheme of one-level basis construction for Harr wavelets in 3D spaces.

With 3D orthogonal base given by $\phi_{kp}^x, \phi_{kp}^y, \phi_{kq}^x, \phi_{kq}^y, \phi_{kr}^z, \phi_{kr}^x$ for bases v_k and w_k , we can have seven wavelets $\phi_{kpqr}^1, \phi_{kpqr}^2, \dots, \phi_{kpqr}^7$, which emphasize details in principal orientations of x -, y -, and z -axes respectively. For example,

$$\phi_{kpqr}^1(x, y, z) = 2^{-3k/2} \phi(2^{-k}x - p)\phi(2^{-k}y - q)\phi(2^{-k}z - r),$$

extracts details in z (depth) direction. Within this filter bank, volume data $A_{2^{n-1}}f$ is filtered along x -, y -, and z -axes into one approximation and seven oriented details $D_{2^n}^1f, \dots, D_{2^n}^7f$ at each step. The filter operations for the three dimensional wavelets is shown in Figure 2.9. The result decomposition data pyramid is shown in Figure 2.10.

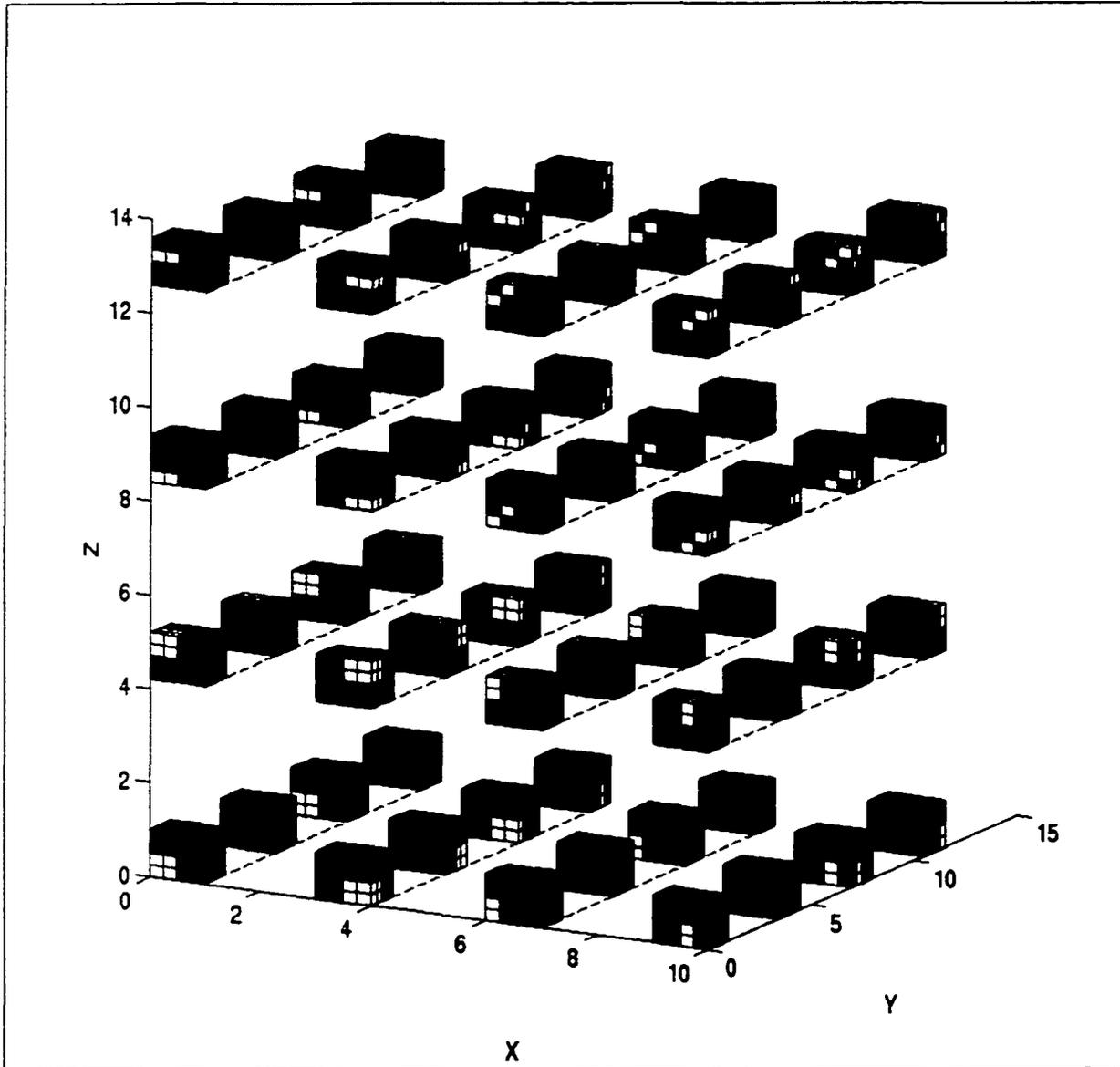


Figure 2.8: The first level 3D basis construction of Harr wavelets. The size of 3D cube is $4 \times 4 \times 4$. White stands for +1, black for -1, and gray for 0.

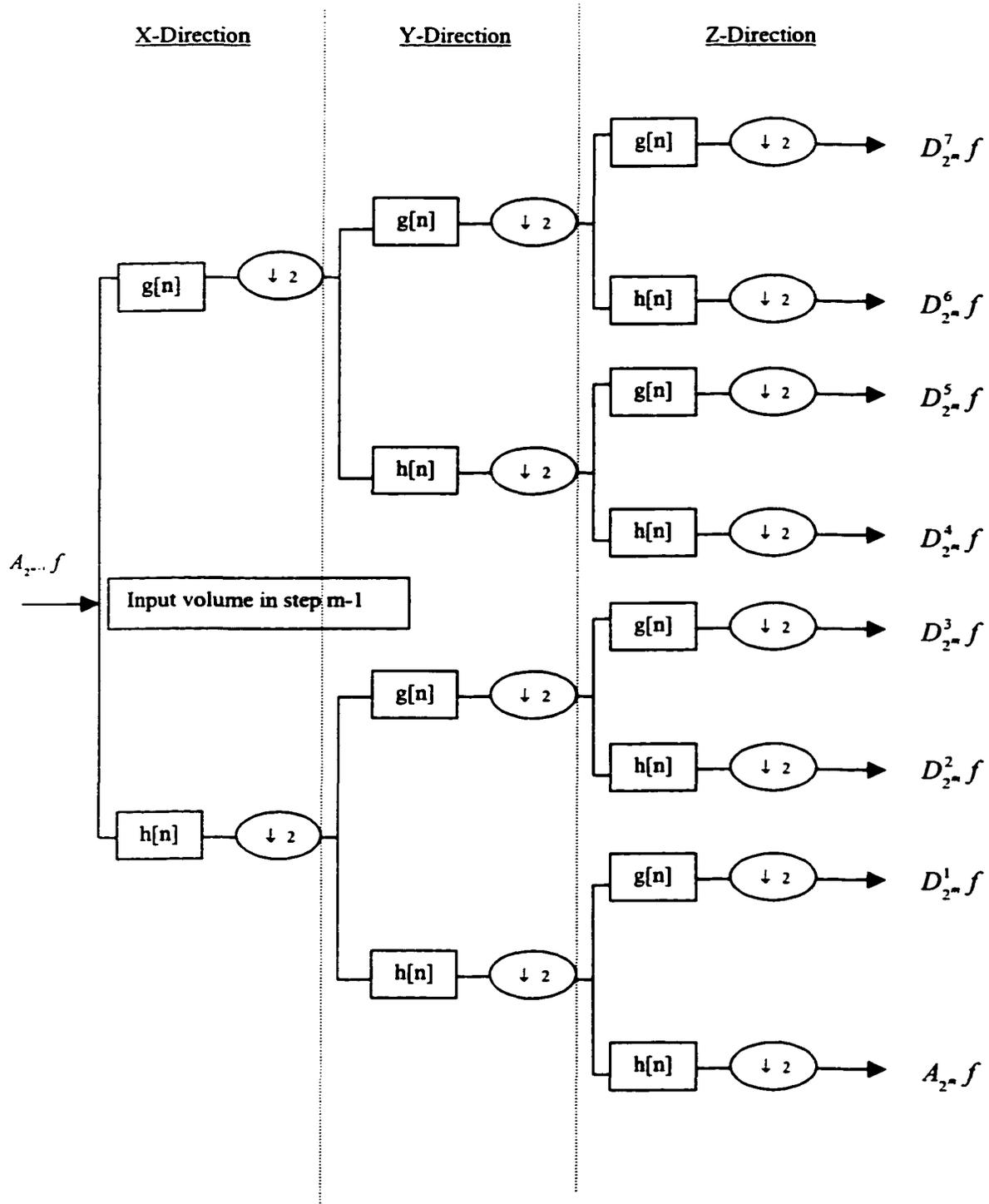


Figure 2.9: Quadrature mirror filter pyramid to implement the 3D-wavelet transform

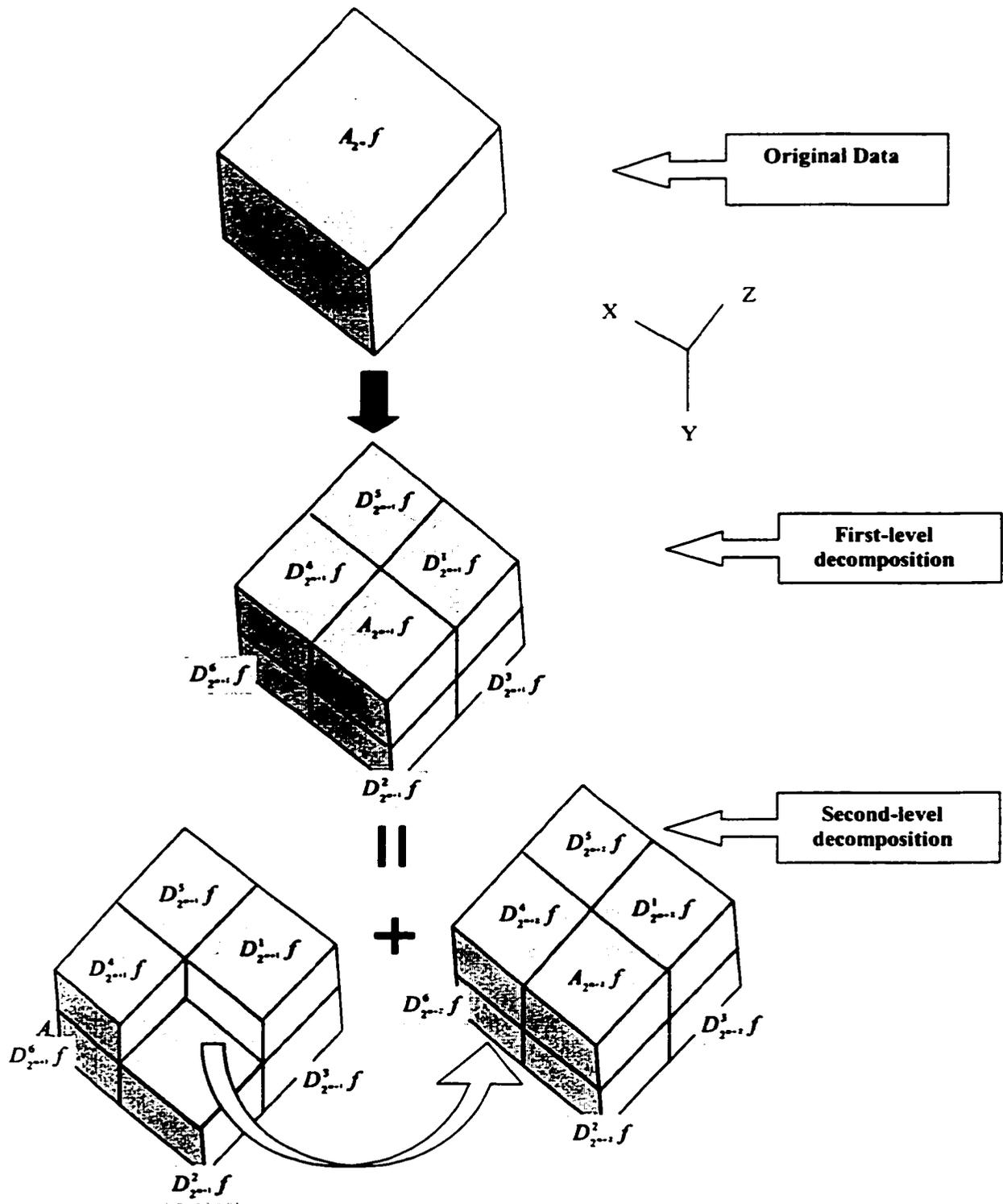


Figure 2.10: Volume data decomposed into eight blocks by 3D wavelets

CHAPTER 3

WAVELET-BASED SURFACE ANALYSIS FOR FORM FEATURE EXTRACTION

3.1 Methodology

The mathematical description of an object by a solid model distinctively determines whether a given point is inside or outside the object volume. In the solid model, surfaces are the boundaries of transition between the inside of the object and its outside. When the solid model is digitized using the cell decomposition and represented by a 3D-binary array, the curvature pattern of a surface is converted into a pattern of curvature singularities in terms of the binary changes between 0 and 1.

In this work, the solid model is first pre-processed by digitizing it into a 3D-binary array. Testing models were chosen from the “National Repository for Feature Recognition, Process, and Assembly Planning”, an effort at Drexel University sponsored by the National Institute of Standards and Technology (NIST). The repository can be accessed by the World Wide Web: <http://edge.mcs.drexel.edu/repository/frameset.html>. The solid model in Figure 3.1 is used to demonstrate surface singularities in the digital solid model. Different singularity patterns of the 3D binary array is visualized in Figure 3.2. Figure 3.3 is the top view of the digitized model and gives a better view of its singularity patterns.

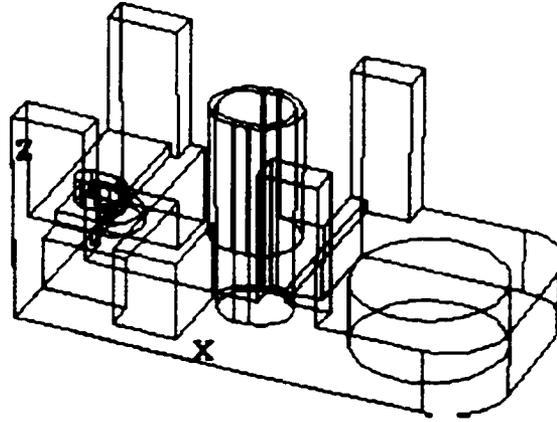


Figure 3.1: Kim's part from NIST design repository

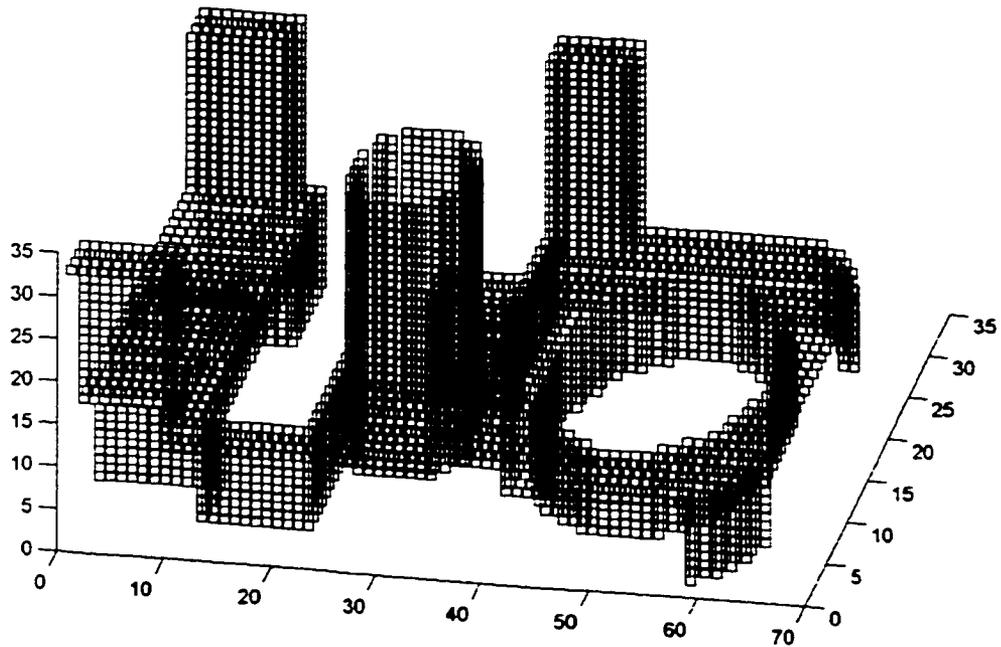


Figure 3.2: Surface singularity patterns in a binary representation

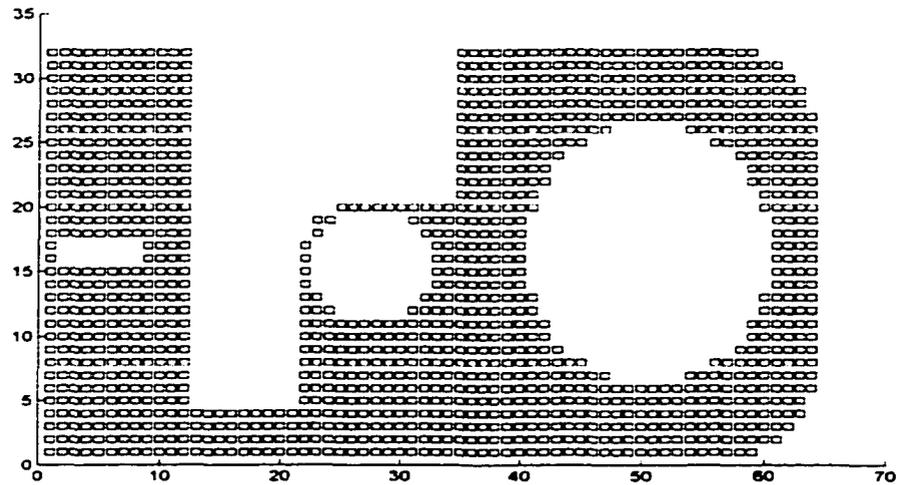


Figure 3.3: Top view of the digitized model in Figure 3.1

The next step is to construct the approximation and detail of the discrete solid model by three-dimensional discrete wavelet transforms (3D-DWT). 3D-DWT can decompose a discrete solid model into an approximation and seven spatially oriented details according to the singularity pattern of its surfaces. Three-dimensional WT can be implemented by tensor products of three 1D Fast Wavelet Transform (FWT). The FWT is equivalent to convoluting the discrete model by two sets of filter coefficients: low-pass filter coefficients \overline{h}_n , and high-pass filter coefficients \overline{g}_n . The filtered data is then down sampled by two. The output of the combination of three low-pass filters is the approximation of the model. The direction-oriented details, which are the output of the spatial combination of the low-pass and high-pass filters, contain spatially oriented singularity patterns of surface curvatures. This is illustrated in Figure 3.4 by the decomposition of the top view image (Figure 3.3). The black and white pixels

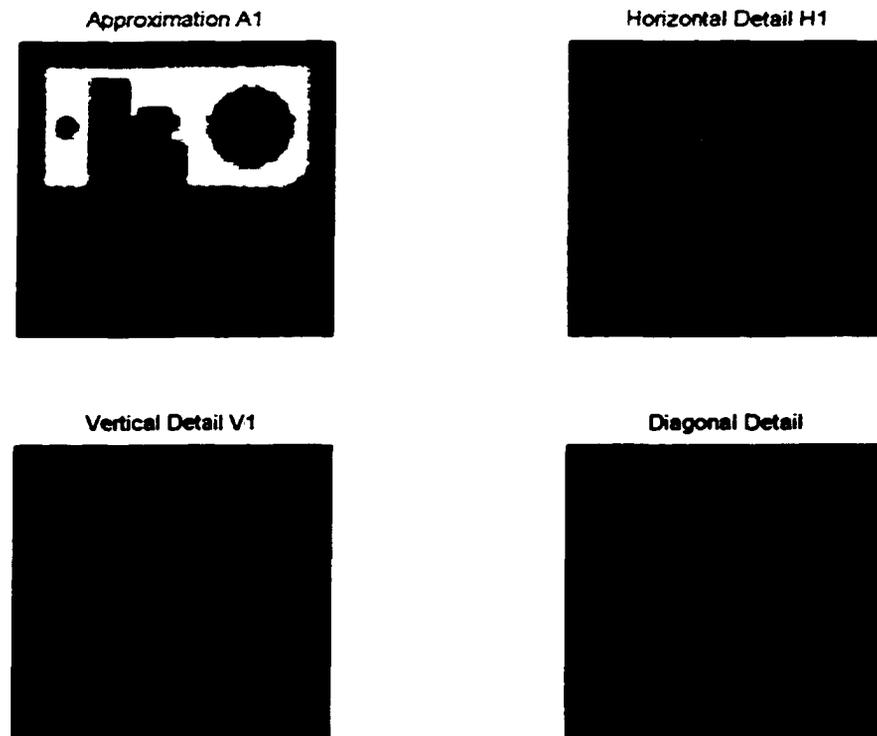


Figure 3.4: One level wavelet decomposition of top view image in Figure 3.3

correspond respectively to zero and high amplitude coefficients. It can be noted that the amplitude is high along the edges for each orientation.

In this decomposition stage, a suitable wavelet base needs to be selected for a “good” performance in distinguishing curvature patterns of surfaces in the solid model. A number of criteria used for choosing the wavelet can be found in the different wavelet applications, such as image coding [Tewfik and Jorgensen, 1991] and computer graphics [Muraki, 1993]. In Tewfik and Jorgensen’s work, a suitable wavelet was considered as one that results in a small number of non-negligible wavelet coefficients or a best approximation of a given signal up to a given scale. Whereas Muraki considered a “good” wavelet function for

volumetric shape descriptions should be either symmetric or anti-symmetric even if the function is not compact.

In this work, the chosen of wavelet is obtained by testing a number of wavelet bases with different vanishing moments on the solid models. In the initial research stage, wavelets with different vanishing moments were applied depending on the complexity of surfaces in a solid model. For a solid model with a variety of details and free form surfaces, a Daubechies wavelet with four vanishing moments is used to obtain information for surface coding and classification. For a solid model with less detail and free form surfaces, Harr wavelet is applied to extract coarse data for coding and classification.

In order to use wavelet coefficients to characterize the surface complexity, the decomposition coefficients are manipulated before wavelet reconstruction is applied. The option chosen for the algorithm is determined by the performance of the reconstructed wavelet coefficients in surface coding and classification. At the initial stage of this research, eight options are proposed and tested for handling wavelet coefficients as listed in Table 3.1. Here \times denotes the deletion operation of the corresponding coefficients and \surd denotes keeping the corresponding coefficients. The energy function is the average of the power of all coefficients.

The discrete solid model is then reconstructed by convoluting the modified wavelet coefficients with two filters. These two filters are generated by reversing the order of the low-pass filter coefficients \overline{h}_n and the high-pass filter coefficients \overline{g}_n respectively. The data is up sampled by two before reconstruction. The reconstructed wavelet coefficients signify the surface curvature singularity information. Since a form feature can be viewed

Table 3.1: Options for wavelet coefficient manipulation before reconstruction

Option	Details	Approximation	Normalization	Filter
1	✓	✓	✗	Energy Function
2	✗	✓	✗	Energy Function
3	✓	✗	✗	Energy Function
4	✓	✓	✓(0 – 100)	Eliminate (0 – 80)
5	✗	✓	✓(0 – 100)	Eliminate (0 – 80)
6	✓	✗	✓(0 – 100)	Eliminate (0 – 80)
7	✗	✓	✗	✗
8	✓	✗	✗	✗

as introducing singularities into the initial discrete solid model, the reconstructed wavelet coefficients with the high magnitude will point to the locations of form features on the model.

When the reconstructed wavelet coefficients are segmented into intervals relative to their maximum, a surface can then be coded and classified in terms of its coefficients' variation pattern. Normally a surface consists of the external contour (loop) and internal content (surface patch) illustrated by Figure 3.5. The internal content of a surface is used in this paper for the surface analysis and representation.

(surface patch) illustrated by Figure 3.5. The internal content of a surface is used in this paper for the surface analysis and representation.

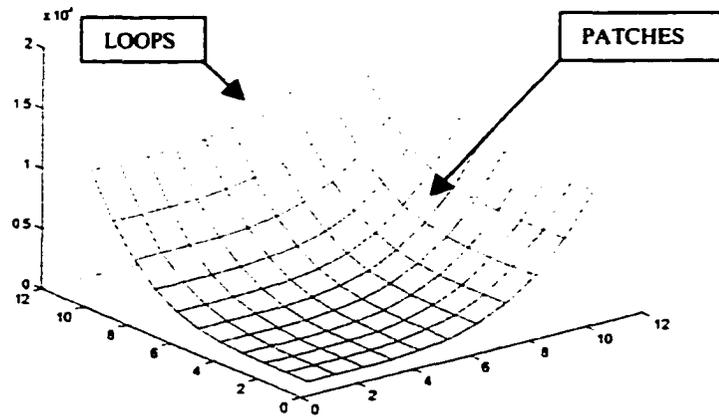


Figure 3.5: A surface with its loops and patches

3.2 Detailed Description of the Surface Ranking Procedure

Models were selected to test the performance of the ranking algorithms on part complexity, interacting features and non-linear surfaces. A graphical user interface (GUI) was developed for user interaction using MOTIF / X Windows and OpenGL toolkits. A snapshot of the GUI can be found in Figure 3.6. In this graphical environment, the user can test different wavelet bases along with different vanishing moments. The user is also able to choose an option to manipulate and threshold decomposition coefficients. The reconstructed object model is displayed on the working window using a color scheme.

Two different experiments were carried out. The first experiment was to study the effect of different wavelet bases and vanishing moments on transform representations. Different levels of wavelet coefficients were investigated in the second experiment.

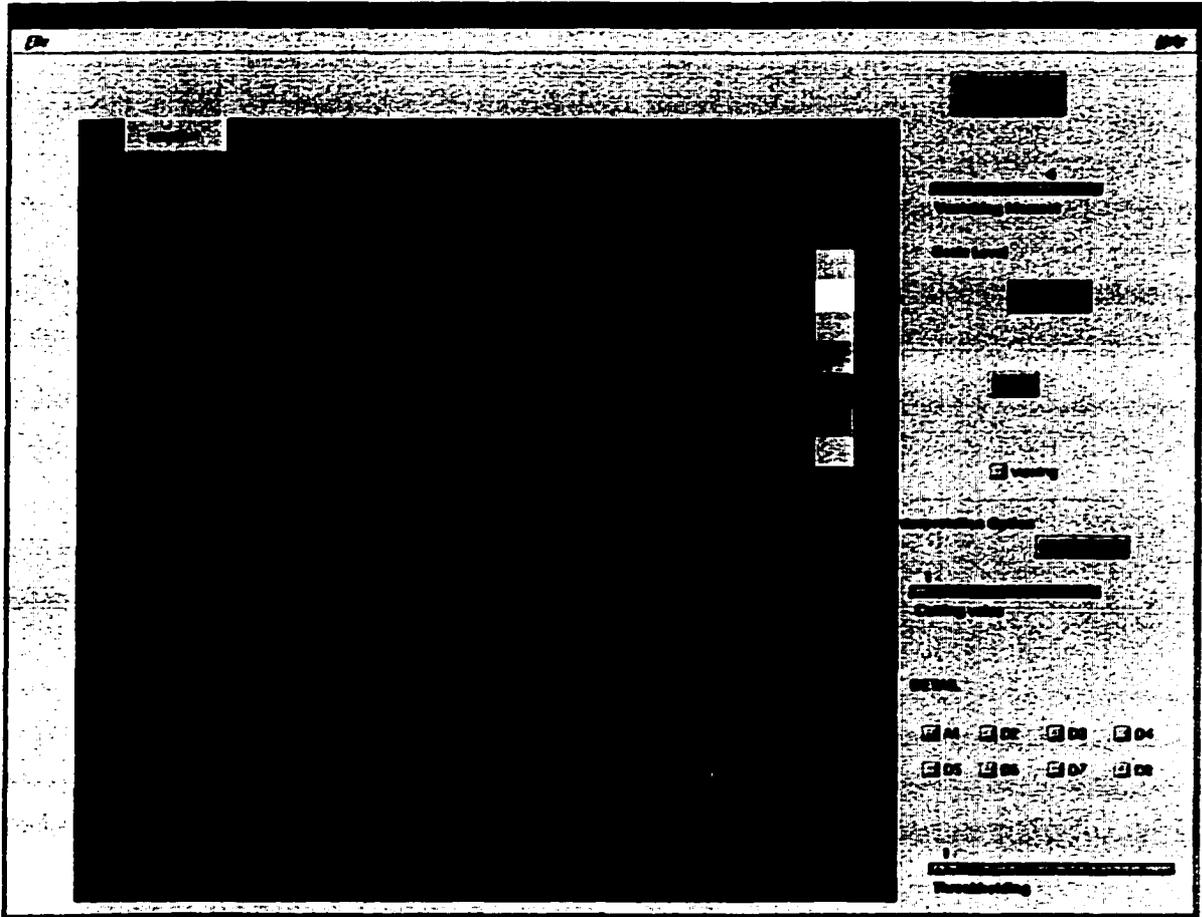


Figure 3.6: A snapshot of GUI interface for surface analysis research

An ACIS part from Spatial Technology was selected as an example in this section and presented in Figure 3.7. This part was chosen due to the presence of free-form features with complex surface curvature properties, in addition to simple features with flat and cylindrical surfaces.

The proposed surface analysis procedure consists of five basic stages:

Stage 1. Digitizing Solid Model

The product model is enclosed in a testing cube of $64 \times 64 \times 64$ and digitized by sample blocks of uniform size ($1 \times 1 \times 1$). The ACIS Test Harness modeling interface is used to model the solid and fit the model into the test cube. ACIS C++ API functions are called to digitize the model. If the block is either fully or partially filled with material, it is labeled

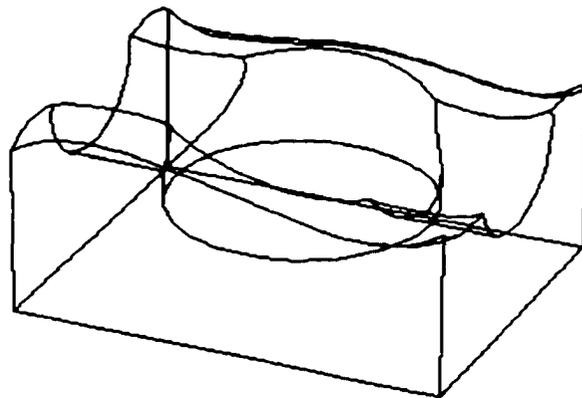


Figure 3.7: An ACIS part with free-form features on its top surfaces

with the value +1; otherwise, it is labeled with the value 0. A “Point-to-Surface” matrix representing the location of points relative to surfaces is also prepared for later point-surface mapping and the calculation of surface point percentage.

Stage 2. Applying Fast Wavelet Transform on the Digitized CAD Model

A 3D-discrete fast wavelet transform, which is implemented by conjugate mirror filters $h[n]$ and $g[n]$, decomposes the discrete model into the approximation and details. This transform is a three-dimensional generation of three 1D-wavelet transforms described above. The 1D-wavelet transform is first applied to each row of the 3D-matrix. It first computes the approximation and detail coefficients in X-direction by taking every other of the convolution of $A_{x...f}$ with $h[n]$ and $g[n]$ respectively. The same 1D operations for the Y- and Z-directions are then applied on the transformed matrix respectively. The filter operations for the three dimensional wavelets is shown in Figure 3.8.

In the implementation of fast wavelet transform on a set of finite length signals of $N \times N$ samples, computations of the convolutions with two filters $h[n]$ and $g[n]$ requires knowing the values of beyond the borders 1 and N . The border problems may be handled with one of the three methods described as follows:

1. Periodizing the signal by reflecting it across the starting and ending points. In this case, the signal becomes periodic with a period of $2N$. This algorithm has the disadvantage of creating large wavelet coefficients at the borders.

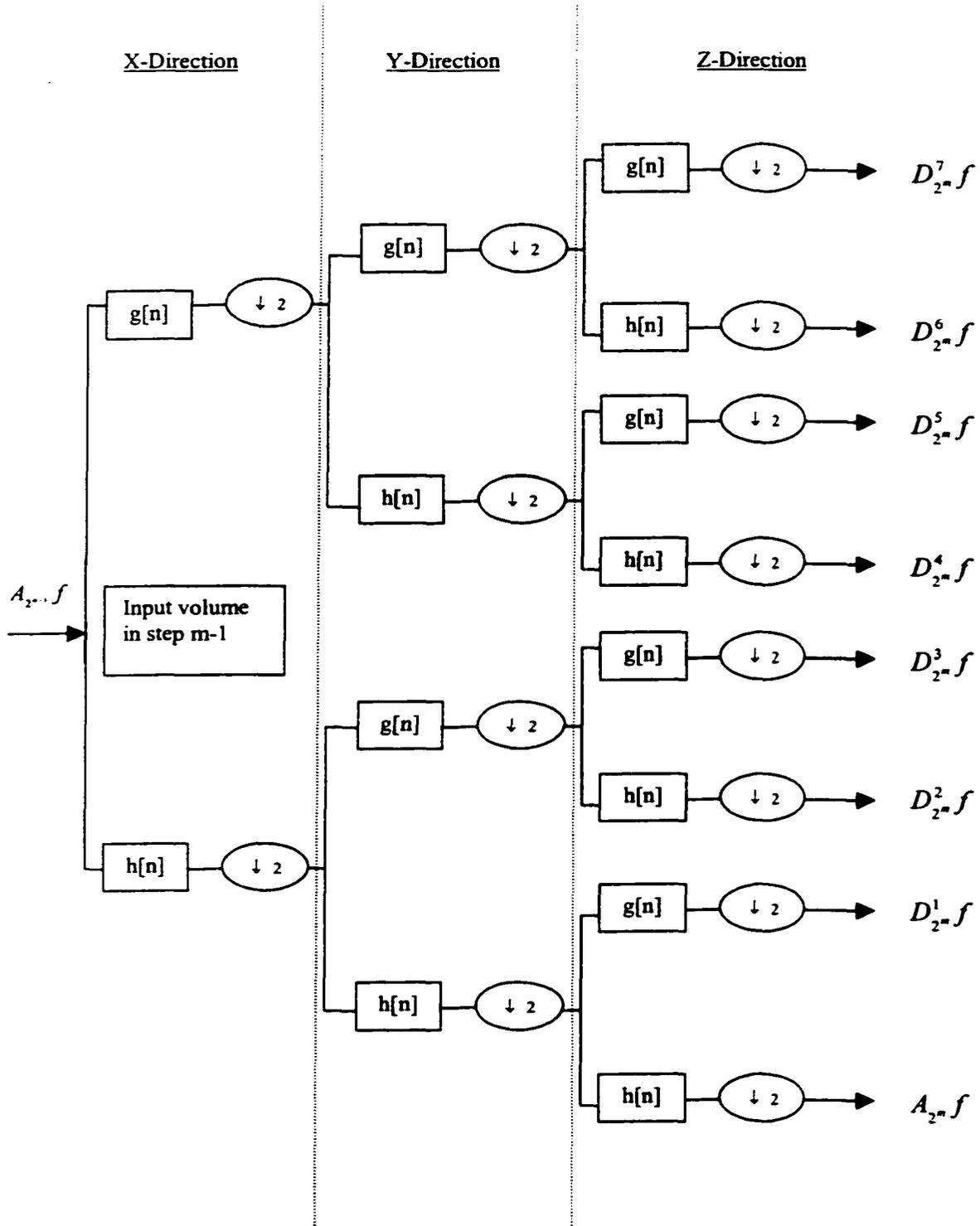


Figure 3.8: Conjugate mirror filter pyramid to implement the 3D-wavelet transform

2. Folding the signal across the start and end points. This algorithm creates smaller wavelet coefficients at the border. However, the wavelet function must be symmetric. As discussed in Chapter 2, Harr is the only symmetric wavelet with a compact support. Spline wavelets have the symmetry but the truncation is needed for numerical calculations.
3. Extending the signal vectors by zeros. This solution will cause additional nonzero coefficients.

Considering all the above factors, the first method will be selected for the surface-ranking algorithm in this work.

Stage 3. Selecting Wavelet Coefficients and Reconstructing Cell Values

The purpose of this step is to keep only those frequency terms that emphasize surface details. Different options of manipulating wavelet coefficients have been proposed and tested in this work (Table 3.1). The numerical experiments show that it is best to retain all terms from seven wavelet bases discussed in Section 2.2.4 in Chapter 2. These seven terms retain high frequencies (low scales) that correspond to a detailed information of surface singularity pattern in the signal. Approximation terms that are projections on the scaling function $\phi_{k_p q r}(x, y, z)$ are deleted. That is, $\frac{1}{8}$ of wavelet coefficients are replaced with zeros as shown in Figure 3.9.

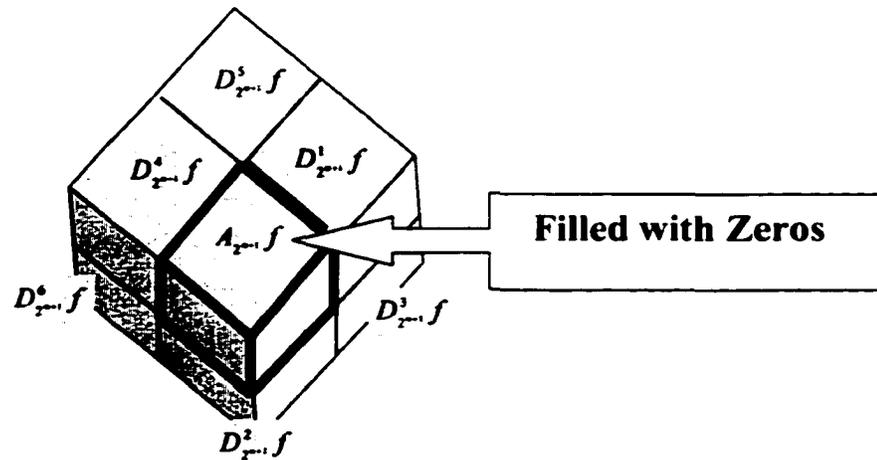


Figure 3.9: Replacing wavelet approximation coefficients $A_{2^{j-1}} f$ with zeros

After approximation terms are deleted, the discrete model is reconstructed by the conjugate mirror filter $h[n]$ and $g[n]$. The reconstruction expands the processed approximation and detail coefficients by inserting zeros between every two of the coefficients and convoluted the expanded coefficients with two filters. The reconstructed representation emphasizes surfaces in various directions, and is ready for ranking and coding.

Stage 4. Segmenting and Thresholding the Reconstructed Values into Surface-Level Representations

This step normalizes the reconstructed cell values by their maximum. Ten intervals, $0\% \rightarrow 10\%$, $10\% \rightarrow 20\%$, ..., $90\% \rightarrow 100\%$, are used for the normalized values' segmentation. For each surface, the distribution of its points over ten intervals is calculated in terms of surface-point percentage.

The surface-level classification is represented by a matrix of order $n \times m$, where n is the total level (here $n=10$) and m is the number of surfaces in the solid model. Harr and Daubechies bases rank the exemplary object presented in Figure 3.7, and the results are

visualized in Figure 3.10 and Figure 3.11 respectively. Table 3.2 lists the surface-level representation using Harr bases. Table 3.3 lists the surface-level representation using Daubechies bases with four vanishing moments. An entity $Surface_Level(i, j)$ in the matrix is the percentage of points on surface j in level i . Numbering for surfaces on the exemplary object can be found in Figure 3.12.

Table 3.2. Surface-level representation by Harr wavelets

Level\Face	1	2	3	4	5	6	7	8	9	10	11
10	0	0.3	0	0	1.2	0	0	1.1	0	0	0
9	5.8	3.9	0	0	4.8	7.9	0	4.1	0	0	0
8	0	0	0.3	0.3	6.2	0	0	6.9	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
6	17	22.5	99.7	99.7	15	28.1	0	15.6	100	0	0
5	0	0	0	0	9.7	0	0	5.3	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
3	26.8	31.6	0	0	6.4	17.1	0	18.6	0	0.1	0
2	0.3	0	0	0	7.6	0	0	3.4	0	0.1	0
1	50.1	42.1	0	0	49.2	47.1	100	45	0	99.8	100

Table 3.3: Surface-level representation by Daubechies wavelets with vanishing moment

Level\Face	1	2	3	4	5	6	7	8	9	10	11
10	0	0	0	0	0.2	0	0	0	0	0	0
9	0	0	0	0	1.2	0	0	0.2	1	0	0
8	0	0	0	0	5	1	0	1.8	0	0	0
7	3.7	4.9	0	0	7.8	12.8	0	8.3	0	0	0
6	3.7	4.6	0	0	12.1	5.9	0.6	11	1	0	0
5	9.3	1.1	0	0	18.5	2.9	98.4	12.6	0	0	93.3
4	28.1	19.3	0	0	17.6	11.9	1	17	0	92.5	6.5
3	17.8	22.8	60.8	19.2	11.6	36.5	0	17.7	11	7.2	0.2
2	13	16.5	30.2	71	16.4	12.2	0	13.8	80	0.3	0
1	24.4	30.9	9	9.8	9.5	16.7	0	17.7	9	0	0

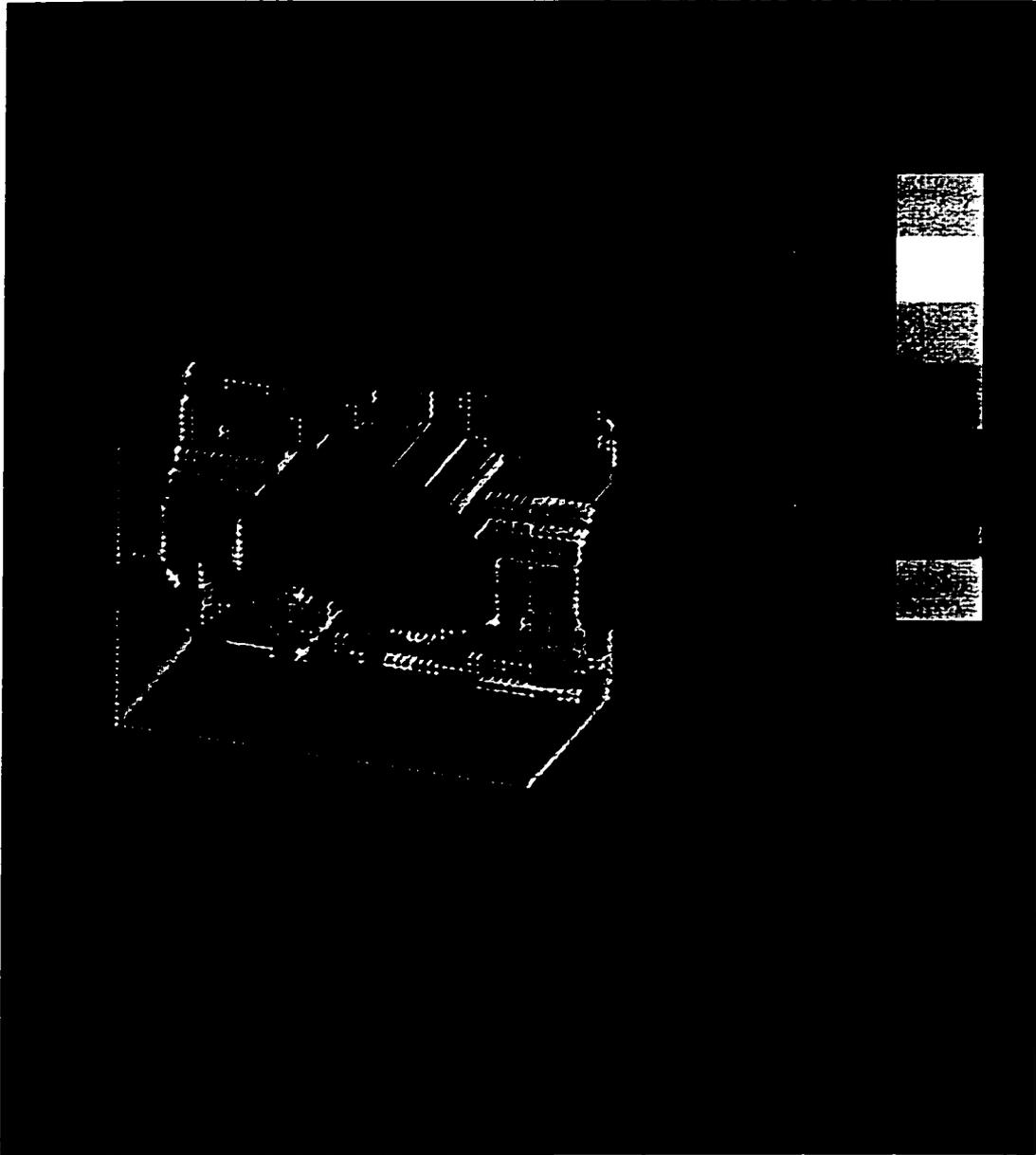


Figure 3.10: Surface levels through the Application of Harr DWTs

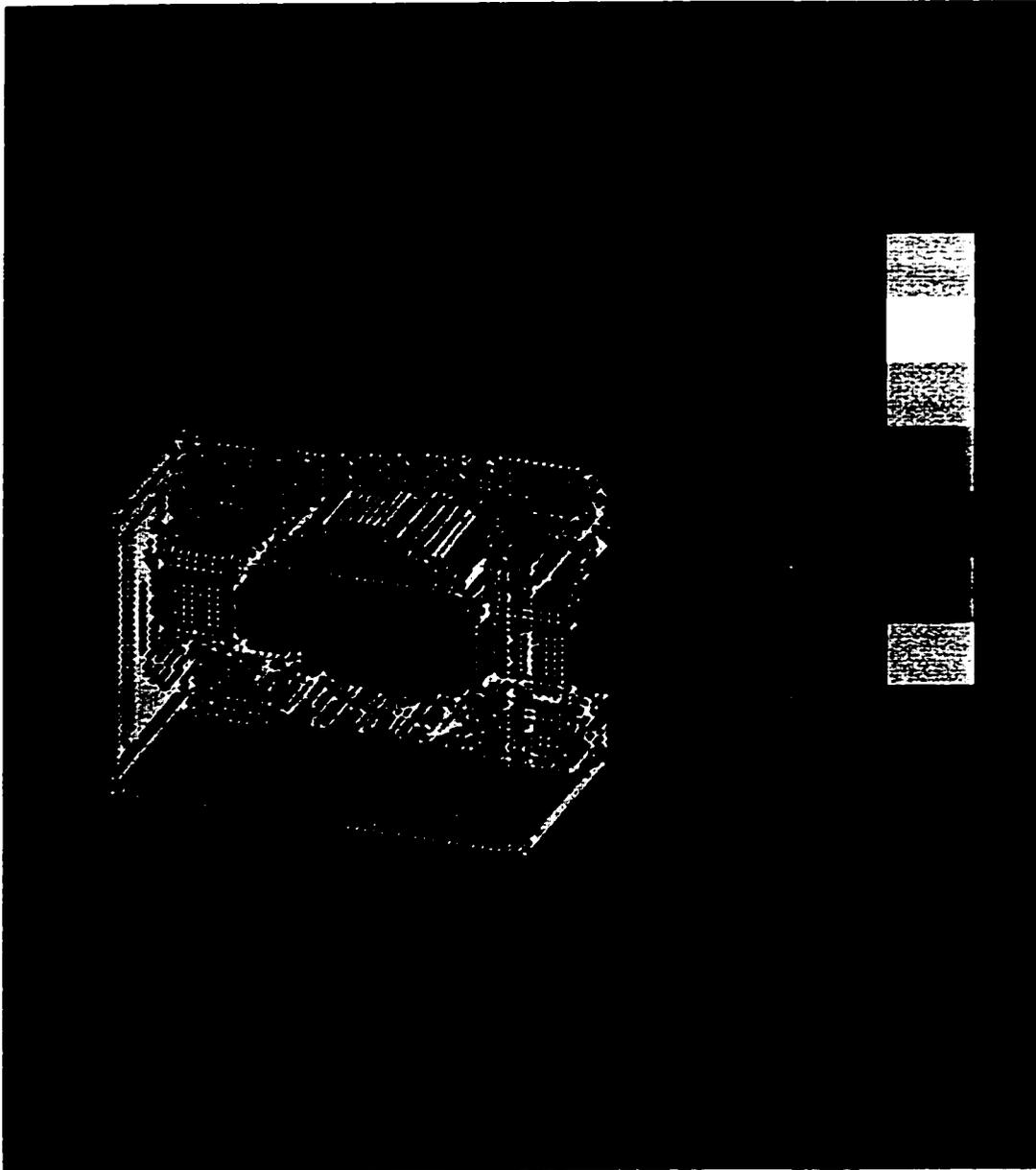


Figure 3.11: Surface levels through the application of Daubechies DWTs
with vanishing moments four

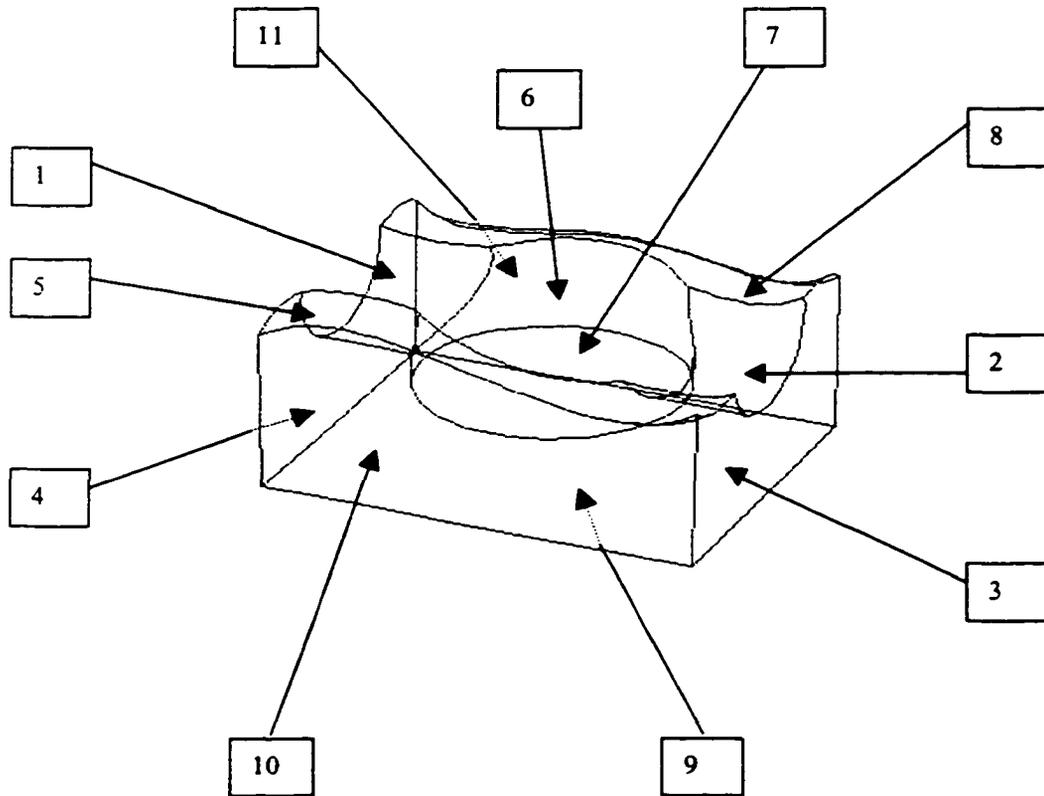


Figure 3.12. Numbered surfaces for the part

While the surface details are noticeable after the reconstructed values are segmented, the surface-level matrix still has some trivial surface-point percentage values that should be removed for the next clustering stage. An evaluation function is then applied on the surface-level matrix to be sure that levels with small surface-point percentage are eliminated.

The evaluation processing is done as follows. First, the matrix is scanned one column at a time for the maximum value of the column that corresponds to a surface. In Table 3.2 and 3.3, it can be noted that flat surfaces paralleling one of the coordinate datum planes, such as

Surfaces 3 and 4, have a maximum value larger than 50%. In order to characterize the pattern of flat surfaces, all elements except the one with the maximum value are set to zeros for the surface with maximum value larger than 50%. Second, for the rest of surfaces each element is divided by the maximum value at the corresponding column. Third, the updated matrix is thresholded by a value of 30%. The above processing can be summarized by the following enhance function:

$$E(\text{Surface_Level}(i, j)) := \begin{cases} 1 \Rightarrow \max(\text{Surface_Level}(i,)) \\ 0 \Rightarrow \text{the rest nine levels} \end{cases} \quad \text{for flat surfaces // datum planes} \quad (3.1)$$

$$\begin{cases} 0 \Rightarrow \text{Surface_Level}(i, j) / \max(\text{Surface_Level}(i,)) < 30\% \end{cases} \quad \text{for other surfaces}$$

Table 3.4 lists the thresholded surface-level representation by Harr bases. Table 3.5 lists the thresholded surface-level representation by Daubechies bases. Table 3.4 and Table 3.5 are plotted in Figure 3.13 and Figure 3.14 respectively. Each column vector in Table 3.4 and 3.5 will be used for surface representation and ranking. As an example, the level representation vector for surface 5 (free form surface on the top of the object) by Debechies bases is [0 0 0 0.42 0.65 1 0.95 0.63 0.89 0.51].

Stage 5. Clustering Surfaces into Form Features

The purpose of this stage is to demonstrate how the surface-level representations obtained from Stage 4 can be utilized for surface grouping into form features. Since the object model in Figure 3.7 has free form surfaces, the Daubechies surface-level representations listed in Table 3.5 are selected for the clustering stage in this example.

Table 3.4: Harr surface level representation thresholded by 30%

Level\Face	1	2	3	4	5	6	7	8	9	10	11
10	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
6	0.34	0.53	1	1	0.3	0.6	0	0.35	1	0	0
5	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
3	0.53	0.75	0	0	0	0.35	0	0.41	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	1	1	1	0	1	1

Table 3.5: Daubechies surface level representation thresholded by 30%

Level\Face	1	2	3	4	5	6	7	8	9	10	11
10	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0.42	0.35	0	0.47	0	0	0
6	0	0	0	0	0.65	0	0	0.62	0	0	0
5	0.33	0	0	0	1	0	1	0.71	0	1	1
4	1	0.62	0	0	0.95	0.33	0	0.96	0	0	0
3	0.63	0.74	1	0	0.63	1	0	1	0	0	0
2	0.46	0.53	0	1	0.89	0.33	0	0.78	1	0	0
1	0.87	1	0	0	0.51	0.46	0	1	0	0	0

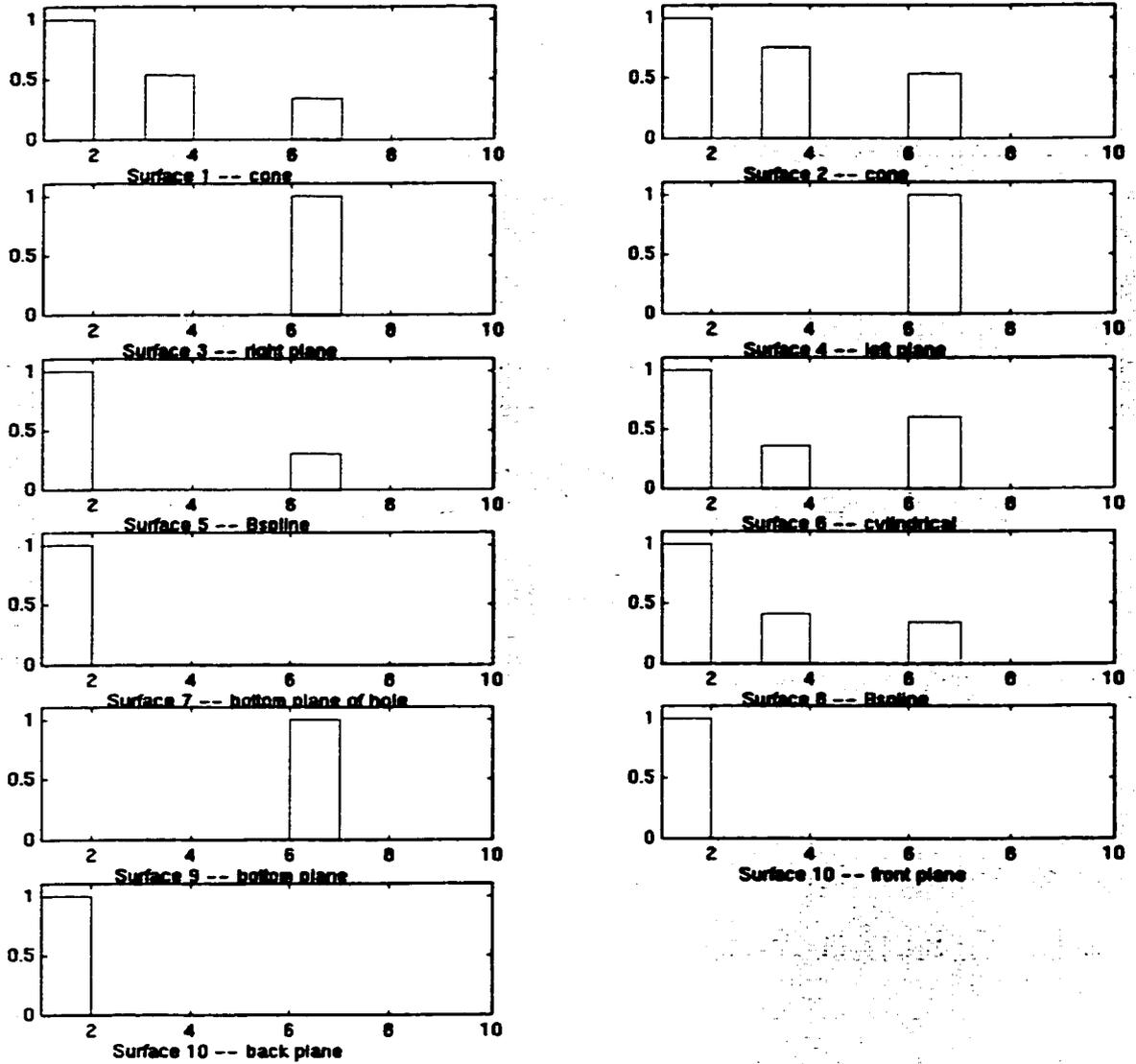


Figure 3.13: Level distribution for surfaces in the part by Harr wavelets

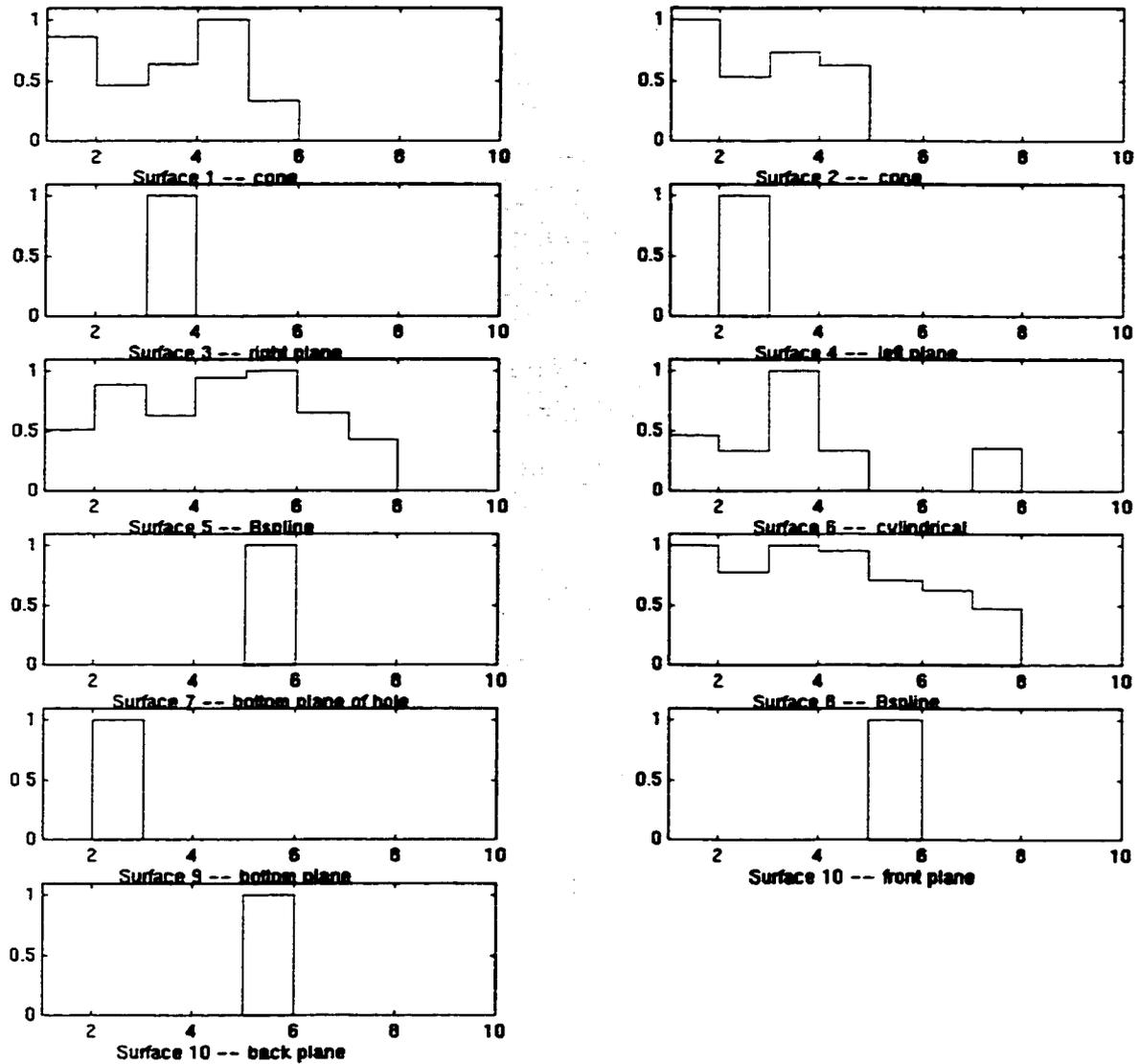


Figure 3.14: Level distribution for surfaces in the part by Daubechies wavelets

The fifth stage of the procedure incrementally generates a graph structure for each surface cluster by scanning each row of surface-level representations. The procedure starts by searching the surface-level matrix with value greater than zero at the highest level (Level 10). Since all surfaces have value 0 from Level 10 to Level 8, no surface has been selected and no surface cluster has been generated till Level 8.

Next, Level 7 is searched where Surfaces 5, 6, and 8 are identified. Once there are surfaces identified, the geometry, topology, and manufacturing context for each of the identified surfaces will be explored. Based on the exploration, the surfaces will either form new clusters or be put into available clusters. In this example, surface connectivity and the convexity of the shared edge of two connected surfaces are considered. As a rule chosen for this example, only surfaces sharing concave edges could be in the same clusters. As a result, three surface clusters (SFC), SFC-1 with Surface 5, SFC-2 with Surface 6 and SFC-3 with Surface 8 as the root node respectively, are generated. The three columns corresponding to the selected surfaces are eliminated from Table 3.5 for later computation efficiency. The updated matrix in Table 3.6 now has the order of 10×8 .

It can be noted that there is no new surface found at Level 6. In Level 5, four surfaces (Surface 1, 7, 10 and 11) are identified simultaneously. Since Surface 7 is connected to Surface 6 by a concave edge, Surface 7 is clustered into SFC-3 as a child of Surface 6. Three more new surface clusters, SFC-4 with Surface 1, SFC-5 with Surface 10, and SFC-6 with Surface 11, are generated. The columns for Surfaces 1, 7, 10 and 11 are then deleted and the matrix now has the order of 10×4 , as presented in Table 3.7.

Table 3.6: Updated level matrix by Daubechies wavelets after Surfaces 5, 6 and 8 are identified

Level\Face	1	2	3	4	7	9	10	11
10	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
5	0.33	0	0	0	1	0	1	1
4	1	0.62	0	0	0	0	0	0
3	0.63	0.74	1	0	0	0	0	0
2	0.46	0.53	0.5	1	0	1	0	0
1	0.87	1	0	0	0	0	0	0

Table 3.7: Updated level matrix by Daubechies wavelets after Surfaces 1, 2, and 6 are identified

Level\Face	2	3	4	9
10	0	0	0	0
9	0	0	0	0
8	0	0	0	0
7	0	0	0	0
6	0	0	0	0
5	0	0	0	0
4	0.62	0	0	0
3	0.74	1	0	0
2	0.53	0.5	1	1
1	1	0	0	0

The procedure is performed recursively on the updated matrix until lowest Level 1 is reached. Eventually all surfaces are added into one of the generated clusters. Figure 3.15 shows the graph structures for surface clusters generated by the above grouping procedure. The shaded circle in Figure 3.15 represents the root node for the corresponding SFC.

An evaluation of the surface-level representations results in the decomposition of surfaces into different clusters. Within each cluster, the root node serves as a “seed surface”, which is used as a common base or starting point for the evaluation of a cluster graph structure. The benefit of using seed surface becomes more significant as the number of surfaces in a model increases.

In order to illustrate the graph structure evaluation, SFC-3 is selected as an example. Here Surface 6 is the seed surface. Surface 7 and seed surface are identified as a single set (blind hole).

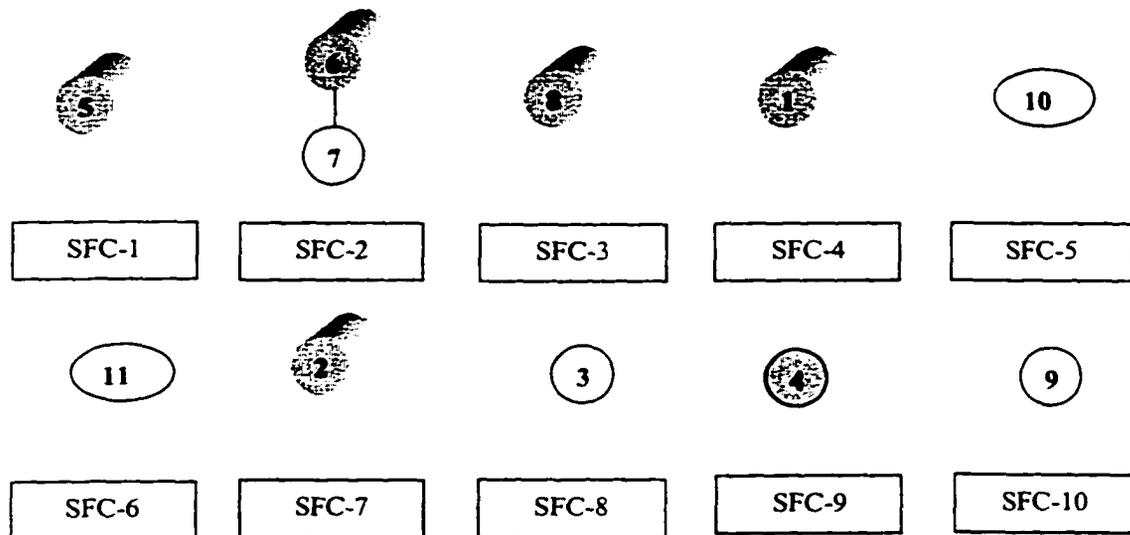


Figure 3.15: Graph structures for surface clusters through the application of Daubechies wavelets

Once the form clustering was obtained as in Figure 3.15, the grouped surfaces could be further classified with additional domain-specific manufacturing information. For this example, machining was chosen as the manufacturing process and two setups were identified for machining the exemplary object. Surface clusters were distinguished by a six-element vector consisting of a machine, a setup, a tool ID, accessibility, machinable surface, and setup surface. The details can be found in Table 3.8.

Table 3.8: Additional manufacturing information for form clusters in Figure 3.15

Form Classification and coding of surfaces	Manufacturing application domain					
	Machining					
	Machine	Setup	Tool ID	Accessibility	Machinable surface	Setup Surface
{5}	CNC machining center	1	/	NO	NO	/
{6, 7}			/	NO	NO	/
{8}			/	NO	NO	/
{1}			/	NO	NO	/
{10}			/	YES	YES	/
{11}			/	YES	YES	/
{2}			/	NO	NO	/
{3}			/	YES	YES	/
{4}				YES	YES	/
{9}			Flat Milling	YES	YES	/
{5}			CNC machining center	2	Ball Milling	YES
{6, 7}	Drilling	YES			YES	/
{8}	Ball Milling	YES			YES	/
{1}	Ball Milling	YES			YES	/
{10}	Flat Milling	YES			YES	/
{11}	Flat Milling	YES			YES	/
{2}	Ball Milling	YES			YES	/
{3}	Flat Milling	YES			YES	/
{4}	Flat Milling	YES			YES	/
{9}	/	NO			NO	YES

With the additional manufacturing information, the second classification of surfaces could be initiated. Surface 1 and Surface 2 are coaxial and could be machined by the same machine (ball milling) at setup 2. Therefore, Surface 1 and Surface 2 were grouped into cluster SFC-4. The same was done for Surface 3, Surface 4, Surface 10 and Surface 11 and a new cluster-SFC-5 was generated. The results are visualized in Figure 3.16.

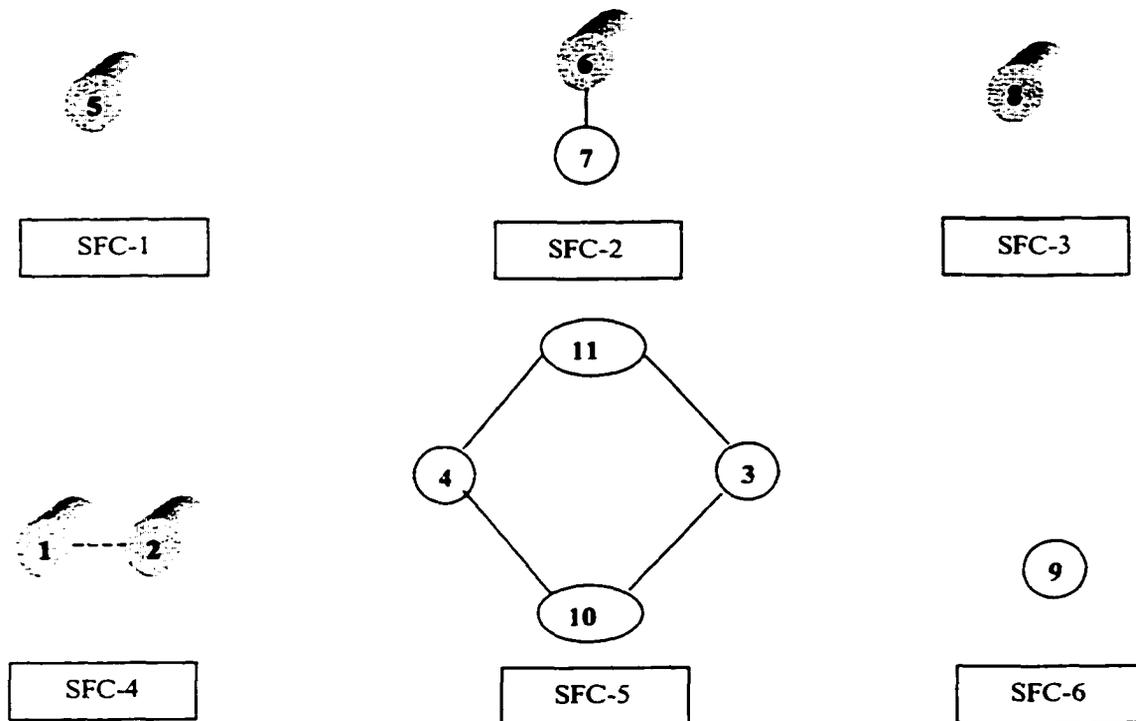


Figure 3.16: Second classification of surfaces in Figure 3.12
with additional manufacturing information

3.3 Discussion of Experimental Results

3.3.1 Studying the Effect of Using Different Wavelet Bases and Vanishing Moments on the Representation

Two wavelet bases, Harr and Daubechies with vanishing moments of 2, 3, 4 and 5 were used for the experiments. The results show that different wavelet bases and the vanishing moments affect the wavelet representation for surface classification and ranking.

The Harr wavelet has a narrow support and is well localized in the spatial domain. The exemplary object in Figure 3.7 is ranked by Harr basis and the result is plotted in Figure 3.13.

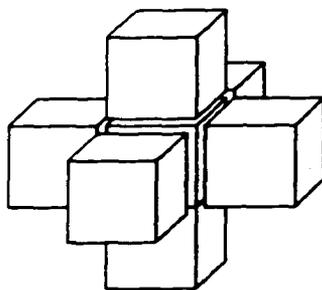
It can be noted that the reconstructed wavelet coefficients cluster in three intervals 50%~60%, 20%~30%, and 0%~10%. This result can be explained by examining the possibility of neighborhood combinations for a cell. Harr wavelet performs the approximation by averaging every two neighbors, so six adjacent neighbors (each 3D-cell has six face neighbors) are involved in its approximation (see Figure 3.16(a)). Harr

conjugate mirror filter h_n is $h(0) = h(1) = \frac{1}{\sqrt{2}}$. Based upon the wavelet theory,

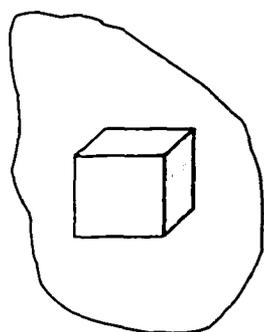
approximation data for a 3D binary representation could be one of the values

$\langle 0, 1, 2, 3, 4, 5, 6, 7, 8 \rangle \times \left(\frac{1}{\sqrt{2}} \right)^3$. These eight values are associated with different geometric

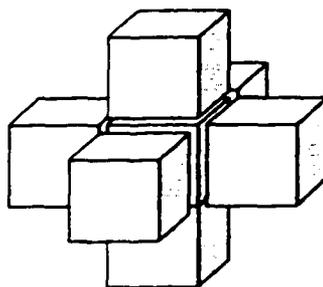
location meanings (see Figure 3.17(b)-(d)). For Example, for points inside a cavity, the approximation value will be 0. While points inside the solid have value 8 and the ones on the flat surface paralleling to one of coordinate datum planes have value 4.



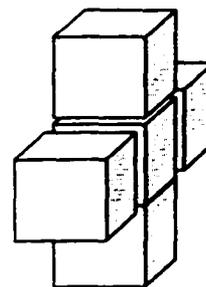
(a) Six neighbors of a 3D Cell in Harr Wavelet representation



(b) Inside a cavity



(c) Inside a solid



(d) On a flat plane

Figure 3.17: A cell's six neighborhoods and its different locations

Daubechies wavelets with two to five vanishing moments were also implemented for constructing the representation. The experiments show that the best vanishing moments of Daubechies bases for surface ranking is four. The results for Daubechies bases are plotted in Figure 3.14. The relationship between the mathematical properties of the wavelet basis and its vanishing moments gives the explanation to this result. Wavelet bases with small vanishing moments have a compact support but a coarse base shape. With the number of vanishing moments increasing, the shape of a base becomes more regular but its support becomes wide. Daubechies with four vanishing moments is a suitable compromise between wavelet shape regularity and wavelet support range.

3.3.2 Studying the Effect of Level of Wavelet Transform on the Representation

Wavelet transform can decompose a signal into a set of approximations and details at different resolutions. When applied to a geometric model, it should act as a multi-resolution window that can represent different geometric details and patterns at different scales.

The wavelet decomposition procedure starts from the first scale $s = 1$ and progresses towards the increasing values of s (low frequencies). The product model in this research is represented by a 3D binary matrix. When wavelet transform is applied to this representation, all singularities will be detected through resolutions. This could be explained by the fact that the only available sharp variation in this binary representation is the one from material to non-material or from non-material to material. The lowest scale (highest frequency) corresponds to a detailed information of a hidden surface singularity patterns in the binary representation. As a result for our binary representation, the first scale is enough to characterize the singularities of surfaces in the product model.

3.4 Summary and Conclusions

Chapter 3 addressed the development of a wavelet-based algorithm generating a surface representation for models and form features. The surface-based representation is proposed as input for the extraction of form features with non-planar surfaces in CAD models. A new shape analysis methodology for form feature extraction and representation by using three-dimensional wavelet transform is presented. The objective of using a wavelet-based shape analysis approach is to overcome the main limitation of the alternative feature extraction approaches, namely their restriction to planar surfaces or simple curved surfaces.

For a free form surface, there is a characteristic pattern of variation in the surface curvatures, which measure the deviation from flatness. A form feature is a region of abrupt change in the shape of a product. As a result, a form feature could be considered as the singularity associated with the “shape signal” of a product. In digital image shape analysis is defined as the process of characterizing shapes and developing shape similarity measures for images. Within the context of CAD models, shape analysis is defined as the process of coding and classification of surfaces.

Three-dimensional discrete wavelet transform (3D-DWT) is applied in this work for the shape analysis of a product model. 3D-DWT can characterize a surface pattern by transforming its spatial information into directional details in different frequency intervals. 3D-DWT uses a prediction method in which the value at a volume cell is approximated by taking a weighed average of its neighboring cells. The detail is the difference between the actual cell value and the approximated value. Detail coefficients with large magnitudes point to the location of the sharp variation in a pattern. Surfaces can then be evaluated by their local performance in terms of wavelet detail coefficients and globally ranked in relative to the maximum.

In the experiments, two wavelet bases (Harr and Daubechies with vanishing moments of two to five) were used for surface analysis. In general, the experiment showed that both wavelet bases work well for CAD models with flat surfaces. When the product model has free form surfaces, the test showed that the Daubechies bases with four vanishing moments construct the best representation. This work also showed that the first level high-frequency decomposition is sufficient for characterizing the singularity patterns in a binary representation of a product model.

A threshold is applied to the surface-level representation before it is used for surface clustering. In each surface cluster, a graph structure is generated incrementally. The graph structure is then evaluated for grouping surfaces into form features. For the purpose of computational efficiency, the term “seed surface”, which has the highest ranking in the clustering graph structure, is defined. The seed surface is used as a starting point for surface grouping into form features. The extracted form feature is a domain independent representation through which a specific application interpretation can be generated.

CHAPTER 4

PARAMETRIC FACE CODING FOR INVARIANT FEATURE REPRESENTATION

4.1 Introduction

The shape variation and contours have rich information for characterizing and representing the geometry of 3D objects [Pavlidis 1982]. In Chapter 3, a column vector of surface-level matrix was proposed as the surface representation and coding at the initial research stage. Nevertheless, this surface-level representation is transformation-variant. The transformation-variance problem could make the feature searching and matching task very difficult as the representation depends on the model's location and orientation.

The reason for the above problem is that discrete wavelet transform (DWT) is unstable under the translation, rotation, and dilation of the input shape signal. Continuous wavelet transform (CWT) can generate translation-invariant representations. It can be written as a convolution product:

$$(CWT_{\phi}x)(u,s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} x(t) \phi \left(\frac{t-u}{s} \right) dt \quad (4.1)$$

Let $x_{\tau}(t) = x(t - \tau)$ be a translation of $x(t)$ by τ , so CWT is translation invariant:

$$(CWT_{\phi}x_{\tau})(u,s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} x(t - \tau) \phi \left(\frac{t-u}{s} \right) dt = (CWT_{\phi}x)(u - \tau, s) \quad (4.2)$$

The DWT is obtained by sampling continuous transforms over a uniform time grid.

Uniform sampling will result in a translation-variant transformation when the samplings of

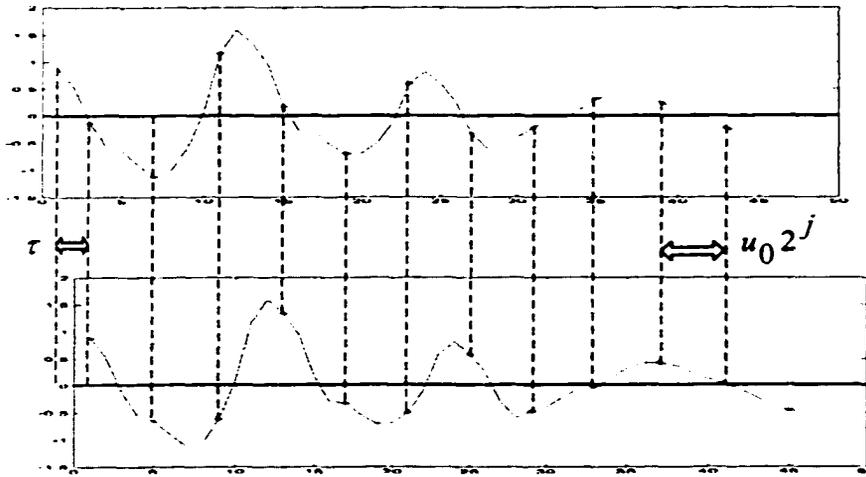


Figure 4.1: Uniformly sampling $(CWT_{\varphi} x_{\tau})(u, s)$ and $(CWT_{\varphi} x)(u, s)$ producing different values if $\tau \neq k u_0 2^j$

$(CWT_{\varphi} x_{\tau})(u, s)$ do not correspond to a translation of the samplings of $(CWT_{\varphi} x)(u, s)$ unless $\tau = k u_0 2^j$, $k \in \mathbb{Z}$, as shown in Figure 4.1.

The main goal of this chapter is to provide transform-invariant representations for CAD models and form features using wavelet transforms. In this work, a face is selected as the unit for model and feature representation. The face is defined as the surface patch and its surrounding contours. Geometrical properties of faces instead of their absolute positions and orientations are considered. Faces can be described in 3D-object space in which a point is expressed by a rectangular coordinate triple (x, y, z) . However, this coordinate representation is unstable under transformation. Alternatively faces can be mapped into a 2D-parametric space in which a point is expressed by a parameter pair (u, v) . The

parametric representation reduces the dimensionality by one and has transformation-invariant property. In this research, each face is described in a 2D-parametric space.

The rest of chapter is organized as follows. Previous work on face characterizations and representations is reviewed in Section 4.2. Section 4.3 gives a brief description of topology and geometry entities in ACIS kernel that is used for geometric modeling representations in this coding research. Section 4.4 reviews the relevant differential geometry for surface curvature definition and computation. Section 4.5 presents the curvature-based face region coding and face contour coding by wavelet zero-crossings. Section 4.6 uses an example to demonstrate how a CAD object is represented by the combination of face region codes and contour codes.

4.2 Previous Work on Face Representation

4.2.1 Curvature-Based Description of Geometric Characteristics of Surfaces

Curvature-based approaches for surface characterization and matching have been used for shape analysis [Besl and Ramesh 1986, Thirion and Gourdon 1996, and Higashi et al. 1997], image analysis and data analysis [Yuile 1989, Eberly et al. 1994, and Eberly 1996], and quality control of free-form surfaces [Beck et. al 1986].

Beck et al. (1986) explored several surface analysis tools for visualizing and inspecting computer-generated parametric surfaces. Besides the standard visualization techniques, such as contour map and high-resolution shaded image, several novel curvature-based methods for detailed inspection of parametric surfaces were also presented. Maps of principal curvatures were used to reveal the variation of the magnitudes and directions of the principal curvatures

across the surface. Lines of curvature were used to show the variation of curvature in the principal directions and the directional flow for the maximum or minimum curvature across the surface.

Besl and Jain (1986) presented a curvature-based descriptor to capture invariant surface information for the early stage of computer vision using range images (or depth maps). The mean curvature κ_m and Gaussian curvature κ_g are identified as the local second order surface characteristics that possess several desirable invariance properties and represent extrinsic and intrinsic surface geometry respectively. The signs of these surface curvatures are used to classify range image surface regions into one of the following eight basic types:

1. Peak critical points: $\kappa_m < 0$ and $\kappa_g > 0$,
2. Ridge critical points: $\kappa_m < 0$ and $\kappa_g = 0$,
3. Saddle ridge critical points: $\kappa_m < 0$ and $\kappa_g < 0$,
4. Minimal critical points: $\kappa_m = 0$ and $\kappa_g < 0$,
5. Saddle valley critical points: $\kappa_m > 0$ and $\kappa_g < 0$,
6. Valley critical points : $\kappa_m > 0$ and $\kappa_g = 0$,
7. Pit critical points: $\kappa_m >$ and $\kappa_g > 0$, and
8. Flat surface points: $\kappa_m = 0$ and $\kappa_g = 0$.

Medioni and Nevatia (1984) presented their shape descriptors in terms of the zero crossings of the Gaussian curvature and the maximum principal curvature, and the local maximum of the maximum principal curvature. Medioni and Nevatia used $\{\kappa_{\max}, \kappa_g\}$ as the pair to contain the exact same surface curvature information as $\{\kappa_{\max}, \kappa_{\min}\}$.

Numerous attempts have been made to construct ridges for the use in image and shape analysis. Ridges are determined by points whose curvatures take a local positive maximum along the specified directions. Various detailed definitions of ridges (and ravines) have been considered in literature. A good survey on the different definitions of ridges can be found in Eberly et al. (1994).

Lang et al. (1997) proposed fast algorithms for the detection of view-independent ridges on surfaces given by graphs of functions. The ridges are defined via extrema of the principal curvatures along the associated principal direction. The approach is based on curvature singularity analysis and allows studying the most important changes of shape on various real world objects, such as peaks of mountains and human facial expression of emotion.

Belyaev and Ohtake (2000) reported the detection algorithm for ridges and ravines on a smooth surface approximated by a triangular mesh. The promising applications of detected ridges and ravines could be in improving mesh decimation techniques and simulating pen-and-ink drawings of 3D objects.

4.2.2 Contour-Oriented Shape Representation and Description

Object contours extracted from images have proved to be useful in object recognition algorithms [Marr 1982, Fan 1987, and Otterloo 1991]. A contour may be parameterized to provide a compact representation of geometric information. Parameterized contours can be described in such forms as arc length, enclosed area, contour tangent slopes, or curvature values. With proper choice of parameterization, contour-oriented shape representations can be invariant to shift, scaling, and rotation.

Van Otterloo (1991) developed mathematical tools to represent the outer boundaries of objects in two-dimensional images. Five information—preserving parametric contour representations that allow for an exact reconstruction of a shape are introduced for use in shape analysis and classification. These are position function, tangent function, acceleration function, cumulative angular function, and curvature function. Based upon parametric contour representations, the measurements of similarity between objects for clustering and classification are provided.

Arbter et al. (1990) reported invariant-matching techniques for 3D objects using Fourier descriptors of their boundaries. A modified arc length parameterization was developed to be linear under an affine transformation and yield the same parameterization independent of the initial representation of the contour. The parameterized boundary description is then transformed to the Fourier domain and normalized there to eliminate dependence on the affine transformation and on the starting point.

Tieng and Boles [1997] presented an affine invariant representation for recognizing planar objects in 3D space based on object contours and the dyadic wavelet transform. Two popular affine invariant parameters, affine arc length and the enclosed area, were chosen for the construction of object representation. Affine arc length can be derived based upon the properties of determinants, and enclosed area is based on the property that all areas are changed in the same ratio under an affine mapping. An arc length parameter involves the second derivative of the contour and is more sensitive to noise than parameters involving only first derivatives or only the bare contour. During the matching procedure, the dissimilarity function defined by extreme wavelet points is used to match objects to models in the database.

Mallat (1991) constructed a multi-scale representation scheme based upon zero-crossings for pattern recognition. The input signal is the contours of the objects resulting from the sharp variation discontinuities of image intensity. Mallat used the second derivative of a smoothing function as the wavelet function in constructing the zero-crossing representations of signals. Zero-crossings of the wavelet transform were used to provide the locations of the signal sharp variation points at different scales.

4.3 Topological and Geometrical Entities in ACIS Modeling Kernel

The work in this chapter is implemented using ACIS 3D Toolkit, which is a product of Spatial Technology Inc. ACIS is an object-oriented C++ library composed of thirty-five Dynamically Linked Libraries (DLLs). It integrates wire-frame, surface, and solid modeling, with both manifold and non-manifold topology. ACIS supports three general types of surfaces: analytic surface, parametric surface, and mesh surface. In ACIS modeling kernel, a face of a solid is a topological entity and is represented as a collection of boundary edges – Contours. The face is a portion of a single underlying geometric surface that can be represented by an equation. Each contour determines a portion of the surface inside the face and a portion outside the face. An illustration of geometric and topological information inside the ACIS Solid Modeling Kernel is given in Figure 4.2.

In ACIS, a parametric surface is a mapping from a 3D-object space into a rectangle within 2D-parametric space as shown in Figure 4.3.

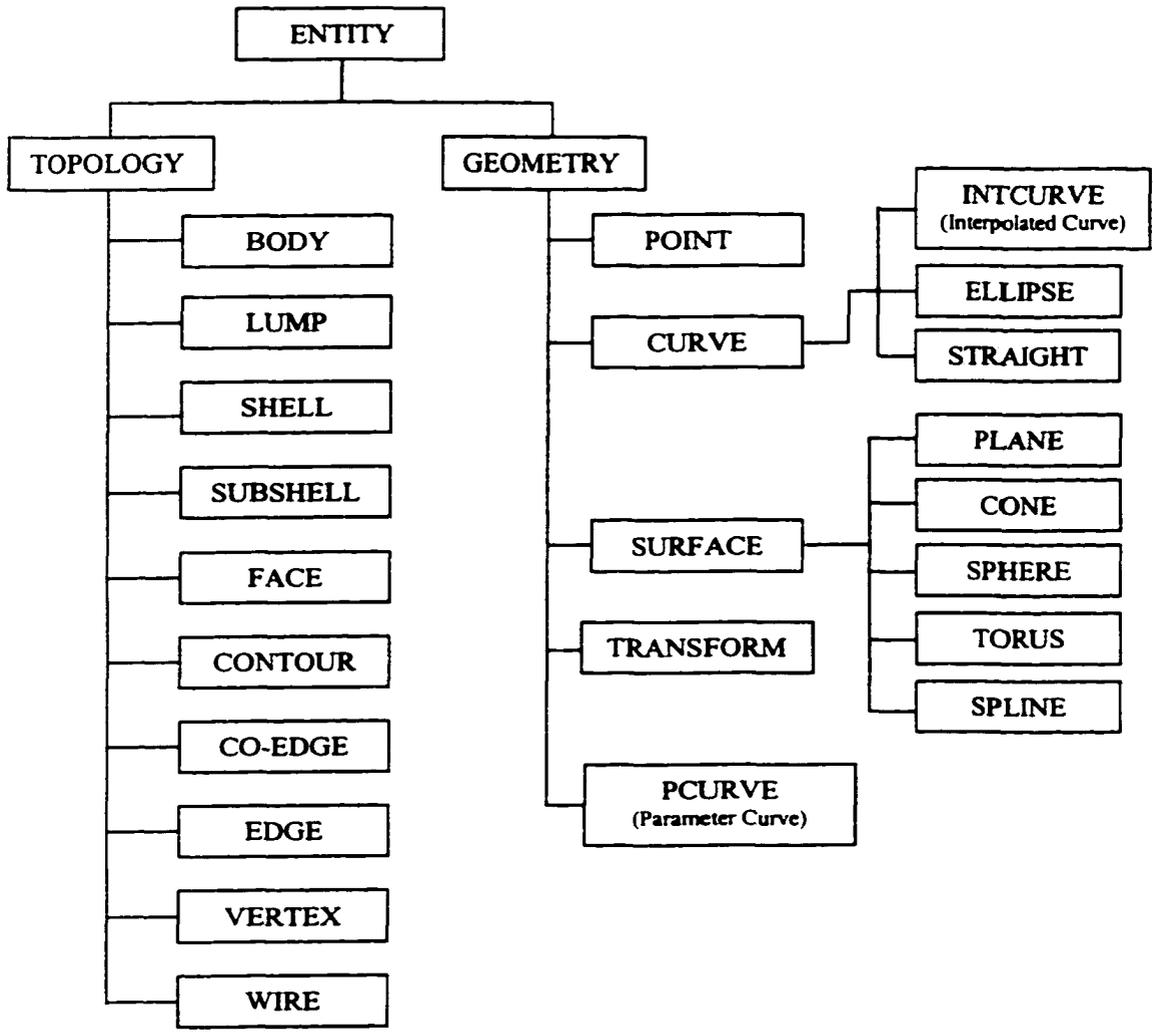


Figure 4.2: Topology and geometry information in ACIS solid molder

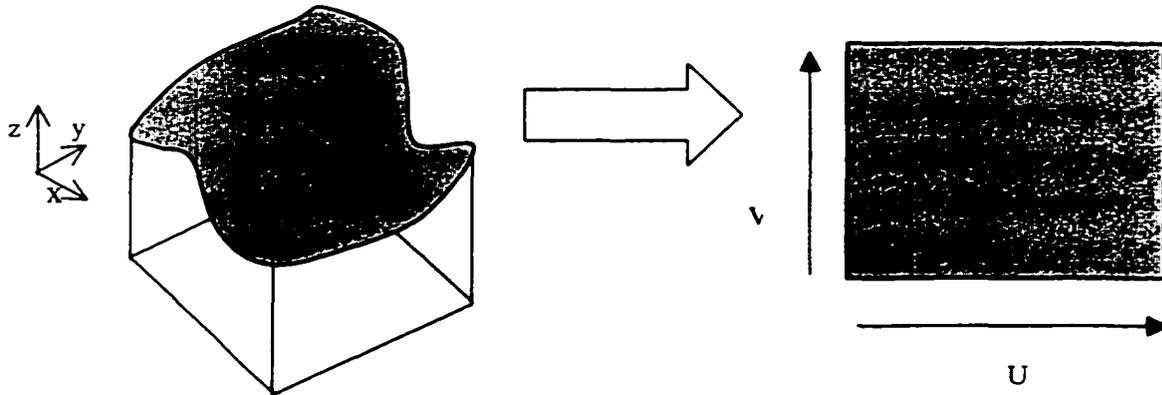


Figure 4.3: Surface mapping

4.4 Surface Differential Geometry

In this section some elementary definitions and results of surfaces and curvatures formulated by classical differential geometry are outlined and discussed.

4.4.1 Surface Normal Curvature

The definition of surface curvature is formulated by differential geometry. Classical differential geometry provides a complete local description of smooth surfaces and plays a central role in the analysis of surfaces.

A regular surface in R^3 is obtained by taking pieces of a plane, deforming them in such a way that the resulting figure has no sharp points, edges, or self-intersections [Hosaka 1992].

The radius of a planar curve at a point P is the radius of the circle that best fits the curve at P . The inverse value of its radius is called the curvature at P . It can also be defined as the

ratio of the counter clockwise rotation angle of the unit tangent vector to the infinitesimal increment ds of the arc length. $\kappa = \frac{d\phi}{ds}$, Figure 4.4.

A normal plane at point P is the plane passing through the point and normal to the surface. The intersection of normal plane and the surface is called the normal section curve C, Figure 4.5. The normal plane is determined by t and n . t is the unit tangent vector of the normal section curve C and n is the unit surface normal vector at point P. The normal section curve is a planar curve and its curvature is called a normal surface curvature at point P.

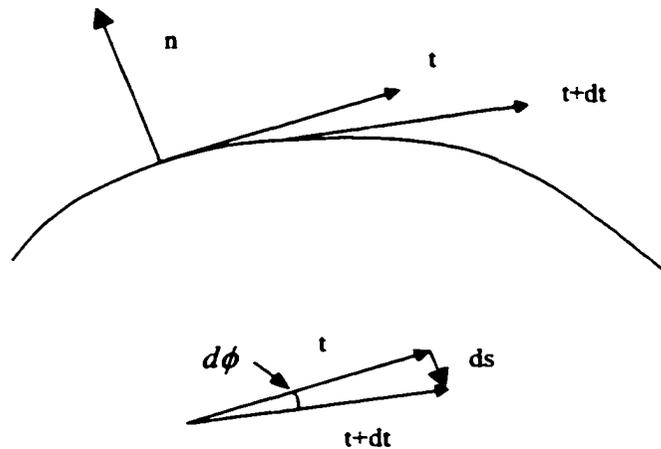


Figure 4.4: Curve curvature definition

4.4.1 Calculation of Surface Curvature

The parametric surface form $[u, v, S(u, v)]$ is directly amenable to a differential analysis, Figure 4.6. The intrinsic differential characteristics of the surface can be derived, such as the principal curvatures and directions, which are independent of parameterization.

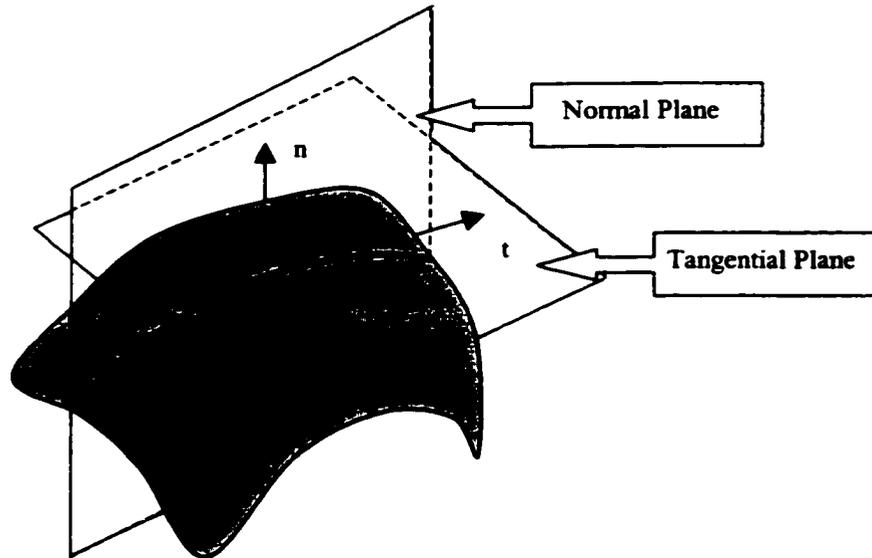


Figure 4.5 A normal curvature of a surface S at a point P —the curvature of a normal section curve C ; t —the unit tangent vector.

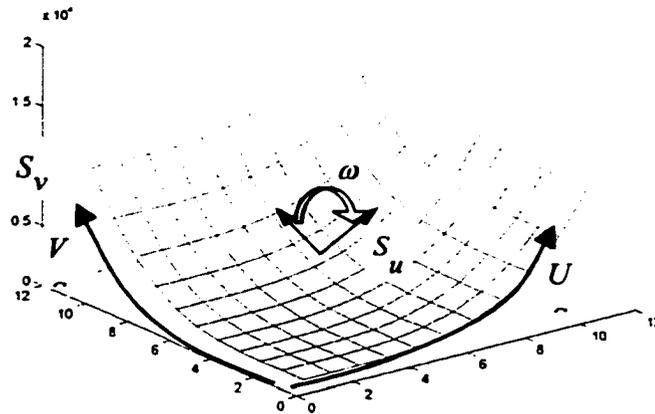


Figure 4.6: Parametric surface patches

A surface patch is denoted in a parametric form by

$$S(u, v) = \{x(u, v), y(u, v), z(u, v)\} \quad (4.3)$$

And

$$dS = S_u du + S_v dv \quad (4.4)$$

So

$$(dS)^2 = E(du)^2 + 2Fdudv + G(dv)^2 \quad (4.5)$$

E , F , and G are the fundamental magnitudes of the first order. They provide a set of differential geometric shape descriptors that capture domain independent surface information.

$$E = S_u^2 \quad (4.6)$$

$$G = S_v^2 \quad (4.7)$$

$$F = S_u \cdot S_v = |S_u| \cdot |S_v| \cos \omega = \sqrt{EG} \cos \omega \quad (4.8)$$

The unit surface normal vector n at point P is then given by:

$$n = \frac{S_u \times S_v}{H} \text{ and } H = |S_u \times S_v| = |S_u| \cdot |S_v| \sin \omega = \sqrt{EG - F^2} \quad (4.9)$$

And the unit tangent vector t at point P is:

$$t = \frac{dS}{ds} = S_u \left(\frac{du}{ds}\right) + S_v \left(\frac{dv}{ds}\right) \text{ where } ds = |dS| \quad (4.10)$$

So

$$\frac{dt}{ds} = S_u \frac{d^2u}{ds^2} + S_v \frac{d^2v}{ds^2} + S_{uu} \left(\frac{du}{ds}\right)^2 + 2S_{uv} \left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) + S_{vv} \left(\frac{dv}{ds}\right)^2 \quad (4.11)$$

And

$$n \cdot \frac{dt}{ds} = n \cdot S_u \frac{d^2u}{ds^2} + n \cdot S_v \frac{d^2v}{ds^2} + n \cdot S_{uu} \left(\frac{du}{ds}\right)^2 + 2n \cdot S_{uv} \left(\frac{du}{ds}\right)\left(\frac{dv}{ds}\right) + n \cdot S_{vv} \left(\frac{dv}{ds}\right)^2 \quad (4.12)$$

Since $\frac{dt}{ds} = \kappa n$,

The curvature κ at P of the normal section curve is called the normal curvature at P,

$$\kappa = \frac{L(du)^2 + 2Mdudv + N(dv)^2}{E(du)^2 + 2Fdudv + G(dv)^2} \quad (4.13)$$

L, M , and N are the fundamental magnitudes of the second order.

$$L = n \cdot S_{uu} = \frac{S_{uu}}{\sqrt{1 + S_u^2 + S_v^2}} \quad (4.14)$$

$$M = n \cdot S_{uv} = \frac{S_{uv}}{\sqrt{1 + S_u^2 + S_v^2}} \quad (4.15)$$

$$\text{and } N = n \cdot S_{vv} = \frac{S_{vv}}{\sqrt{1 + S_u^2 + S_v^2}} \quad (4.16)$$

For many applications, the principal curvatures are of primary interest. Principal curvatures are the extreme values of the normal curvatures in Equation (4.13). Now take

$\gamma = \frac{dv}{du}$ and Equation (4.13) becomes

$$\{L - \kappa(\gamma) \cdot E\} + 2\{M - \kappa(\gamma) \cdot F\}\gamma + \{N - \kappa(\gamma) \cdot G\}\gamma^2 = 0 \quad (4.17)$$

Principal curvatures are obtained by rotating the normal plane and finding the maximum and minimum values of $\kappa(\gamma)$. Principal curvatures are determined by $\frac{d\kappa(\gamma)}{d\gamma} = 0$, which

is:

$$\{EG - F^2\}\kappa^2 - (EN + LG - 2MF)\kappa + LM - M^2 = 0 \quad (4.18)$$

Let κ_{\max} and κ_{\min} be two solutions of Equation (4.18). κ_{\max} and κ_{\min} are the principal curvatures of the surface. The mean curvature κ_m and the Gaussian curvature κ_g can be computed as follows:

$$\kappa_m = \frac{1}{2}(\kappa_{\max} + \kappa_{\min}) = \frac{1}{2} \frac{(EN + LG - 2MF)}{(EG - F^2)} \quad (4.19)$$

$$\kappa_g = \kappa_{\max}\kappa_{\min} = \frac{(LN - M^2)}{(EG - F^2)} \quad (4.20)$$

Finally principal curvatures can be expressed in terms of κ_m and κ_g :

$$\kappa_{\max} = \kappa_m + \sqrt{(\kappa_m^2 - \kappa_g)} \quad (4.21)$$

$$\text{and } \kappa_{\min} = \kappa_m - \sqrt{(\kappa_m^2 - \kappa_g)} \quad (4.22)$$

4.5 The Parametric Face Coding Procedure

A face consists of the internal content (surface patch) and the external closed contour (contour). Both the internal region and external boundary are chosen for face representation and coding. The internal region is characterized by its scalar quantity—curvature distributions. The curvature-based region representations have the following invariance properties [Porteous 1994]: (1) invariance under changes in (u, v) parameterization and (2) invariance under 3D translation and 3D rotations. The contour representation is approximated by the distance between points on the contour and an appropriate reference

position, which is chosen as the center of the region in the parametric space for this research. This distance representation reduces the boundary representation to a 1D function.

Once parameterized representations of face internal region and external contour are obtained, the representations are analyzed in terms of wavelet detail coefficients at different scales. Two-dimensional discrete wavelet transforms (2D DWTS) are applied in this work for the face representation and coding. 2D DWTS can decompose an image into the following set of independent, spatially oriented frequency intervals: (1) low approximation frequencies, (2) vertical high frequencies, (3) horizontal high frequencies, and (4) diagonal high frequencies. The low approximation frequencies can be further decomposed in the next scale.

An ACIS model in Figure 4.7 is used to demonstrate the coding procedure. The model is created by ACIS Test Harness Command—Wiggle.

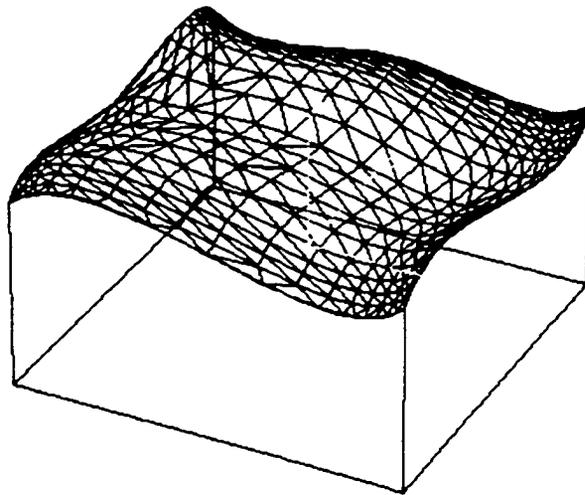


Figure 4.7: An ACIS wiggle model

4.5.1 Wavelet-Based Face Region Coding

In this section, a general strategy is proposed for constructing face region coding algorithms. The face internal region is first characterized by its curvature distributions in parametric space. The proposed wavelet techniques are then applied to decompose the curvature representation into several local coefficients at different resolution levels. The coefficients at each resolution level preserve spatial-oriented details in a limited bandwidth.

The majority of wavelet decomposition coefficients are near zero and a few larger coefficients carry most of the information about the signal. The reason is that the primary goal of designing a wavelet base is to optimize it to produce a maximum number of wavelet coefficients that are close to zero [Mallat 1998]. Considering these facts, a new face representation and coding system is developed based on a few significant points, namely extreme at several resolution levels.

The following are the steps for the region coding procedure.

Step I. Face Region Curvature Computation

Either the principal curvature pair $\{k_{\max}, k_{\min}\}$ or the Gaussian-Mean pair $\{k_g, k_m\}$ can be chosen for the curvature-based shape representation. They contain the exact same surface curvature information. In this work, the principal curvature pair was selected for shape description.

Face regions in parametric space are then sampled on a regular grid. The grid size is selected as a power-two integer. In this case, it is $64 * 64$ points. The sampling processing is implemented by the ACIS 3D Toolkit and its C++ API functions. For each sampling point, its principal curvature pair is evaluated and stored in a two-dimensional matrix. Gray-

level image in Figure 4.8 depicts maximum curvature distribution for the top spline surface in Figure 4.7. For the rest of the chapter, only the maximum principal curvature is used for the demonstration.

Step II. Wavelet Decomposition

Mallat (1989) generalized a multi-scale pyramid algorithm for projecting a two-dimensional array into a set of 2D orthogonal wavelet bases. It decomposes a signal into four spatially oriented components: smoothing components for lowest frequencies in both directions, horizontal high frequency components, vertical high frequency components, and diagonal components for high frequencies in both directions.

Matlab Wavelet Toolbox is used for analysis and synthesis of images using wavelets. A two-dimensional DAUB4 discrete wavelet base is chosen for this experiment. It is implemented by first applying the 1D DAUB4 wavelet transform on each row of the 2D matrix. The same 1D operations are then applied on each column of the transformed matrix.

The biggest number of resolution levels at which a discrete signal can be decomposed is determined by the length of that signal. Most of the energy of the wavelet representation is concentrated on a few coefficients of several resolution levels. Only a few low-resolution levels (highest frequency bandwidths) will be used for distinguishing among different curvature representations. In this experiment, the first three resolution levels are used. The first, second, and third level decomposition of the curvature map in Figure 4.8 are shown in Figures 4.9, 4.10, and 4.11 respectively.

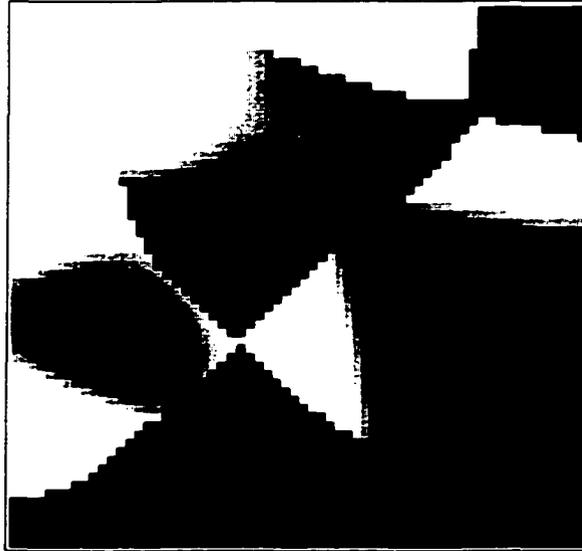


Figure 4.8: Curvature map for the top spline surface in Figure 4.7

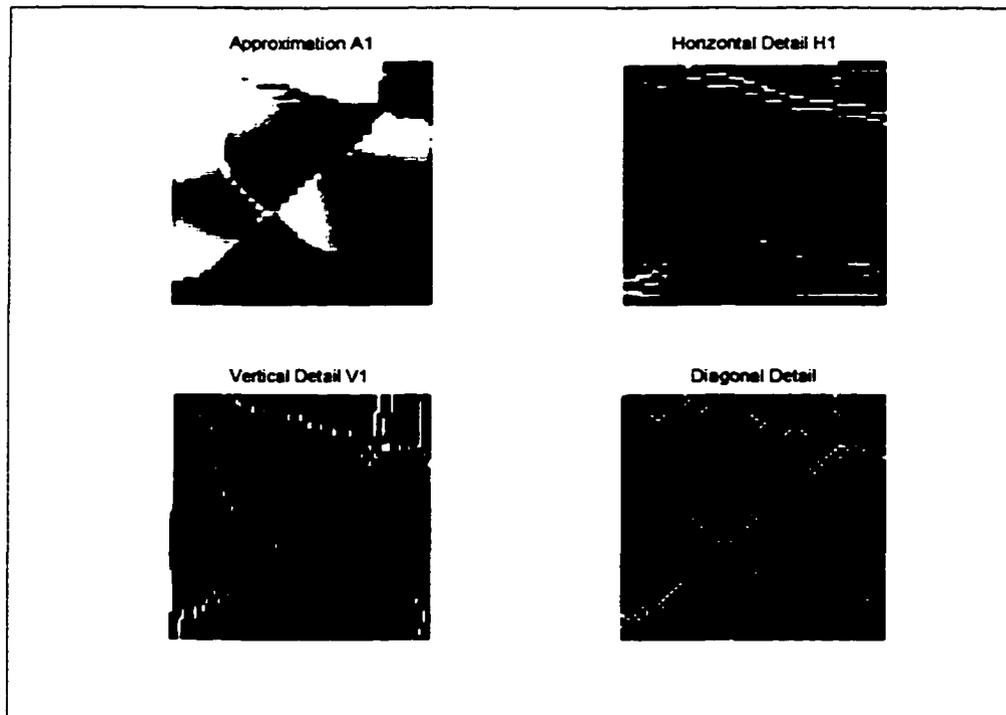


Figure 4.9: The first level decomposition of Figure 4.8

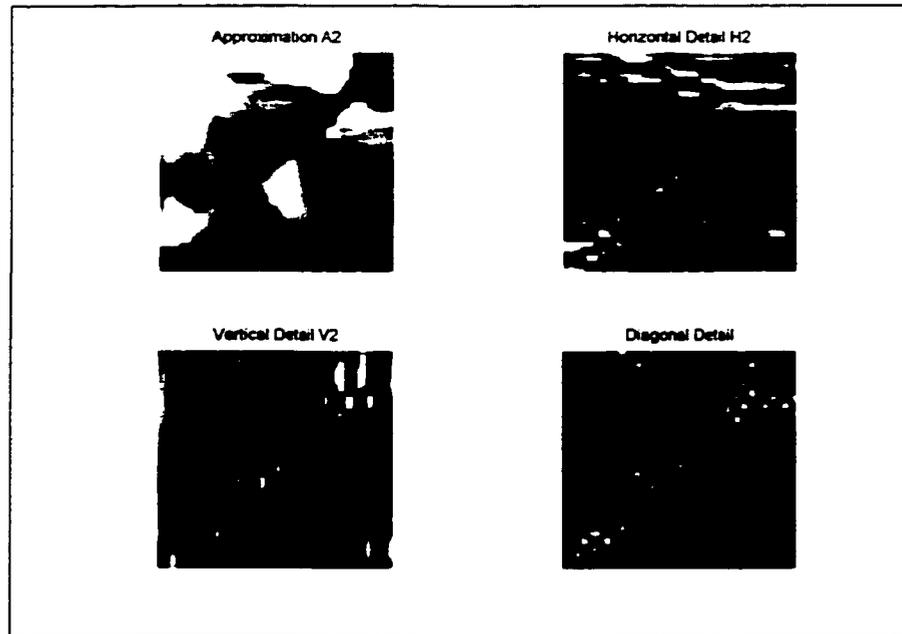


Figure 4.10: The second level decomposition of Figure 4.8

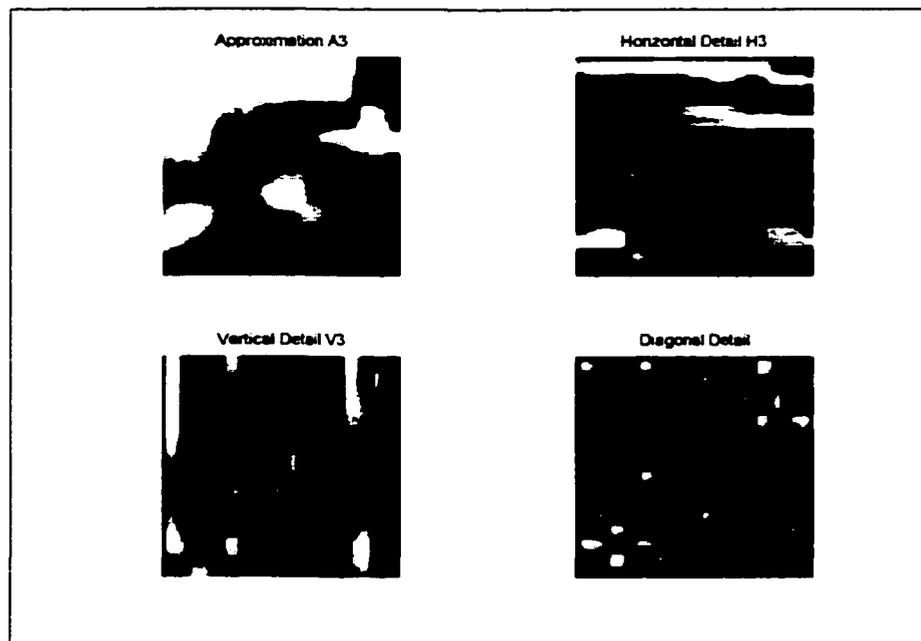


Figure 4.11: The third level decomposition of Figure 4.8

Step III. Multiple-Level Curvature Representation from Wavelet Decomposition

Assume X is a one-dimensional signal, LX is the length of X , and LF is the length of filters. CA is the approximation component, and CHD , CVD , and CDD stand for horizontal, vertical and diagonal components respectively. In Matlab Wavelet Toolbox, the lengths of CA , CHD , CVD , and CDD are the same and equal to $FLOOR(LX + LF - 1)/2$. Here $FLOOR(x)$ is a rounding function that returns the largest integer not greater than the argument x . For this experiment, length of approximations and details are depicted in Figure 4.12.

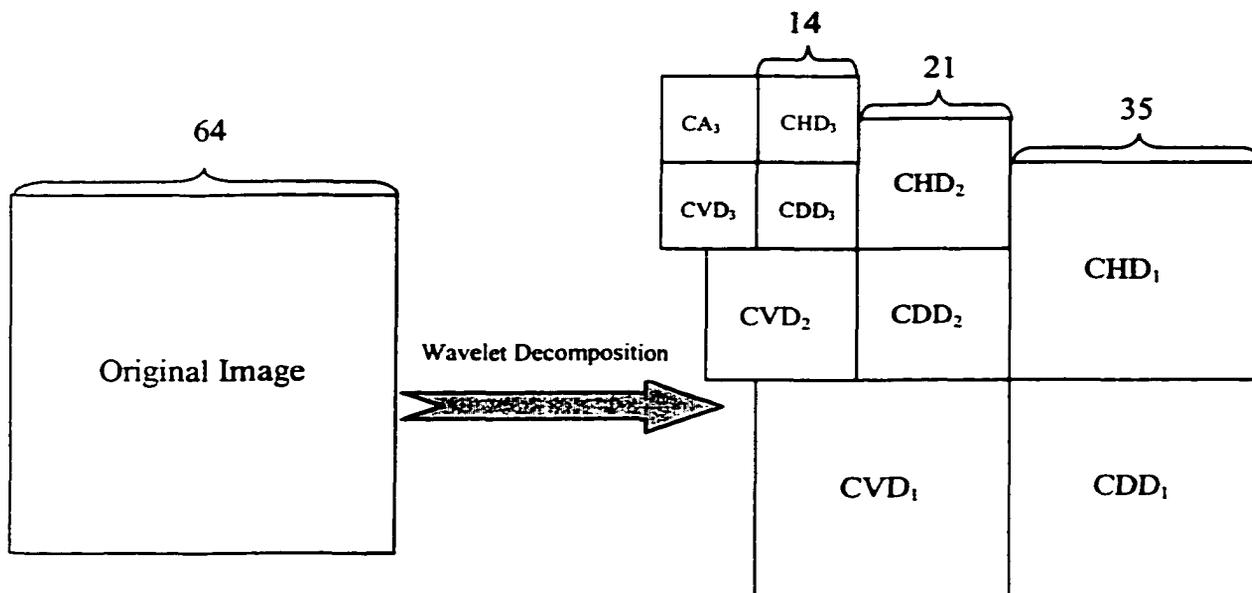


Figure 4.12: Three-level DAUB4 wavelet decomposition in Matlab Wavelet Toolbox

Most of the decomposed coefficients need to be rejected in order for wavelet decompositions of curvature maps to be used for face region representations. For this purpose, a comparison procedure for decomposed wavelet coefficients is developed to search for the most dominant coefficients.

On level one, a comparison among three detail components CHD_1 , CVD_1 , and CDD_1 is made for every point location in the decomposition matrix. The coefficient with the maximum absolute value is then chosen as the dominant coefficient and stored in the corresponding location of a new matrix called “maximum detail matrix”. The maximum detail matrix has the same size as the decomposition matrix. Once the maximum detail matrix is generated, its coefficients are sorted in a descending order and the first twenty-five largest coefficients with their locations are selected as codes for level one. Twenty-five points corresponding to the sorted coefficients are denoted by $P_1, P_2, P_3, \dots, P_{25}$. In order to maintain the invariance to parameterization, the location is represented by the pair $\langle d, \theta \rangle$ instead of $\langle u, v \rangle$. By these twenty-five coefficients, 85% energy of the original coefficients at level one has been retained. Here d is the distance between a point P_i ($i = 2, 3, \dots, 25$) and the largest coefficient point P_1 . θ is the angle from $\overline{P_1 P_2}$ to $\overline{P_1 P_j}$ ($j = 3, 4, \dots, 25$) in counter-clockwise direction, as illustrated in Figure 4.13. The process is done recursively for successive levels until codes for all three-levels are achieved. Figure 4.14 illustrates three-level curvature representations extracted from the decomposition coefficients for the top spline face in Figure 4.6. Its three-level curvature-based codes are listed in Table 4.1.

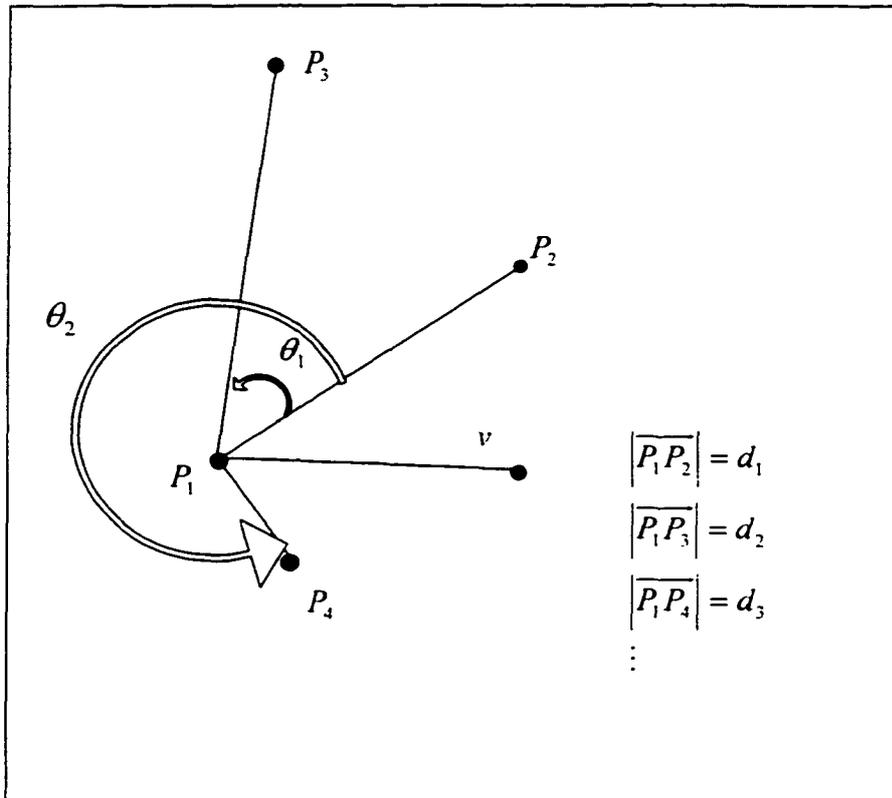


Figure 4.13: Location (d, θ) of a point P in relative to the point M with the maximum coefficient

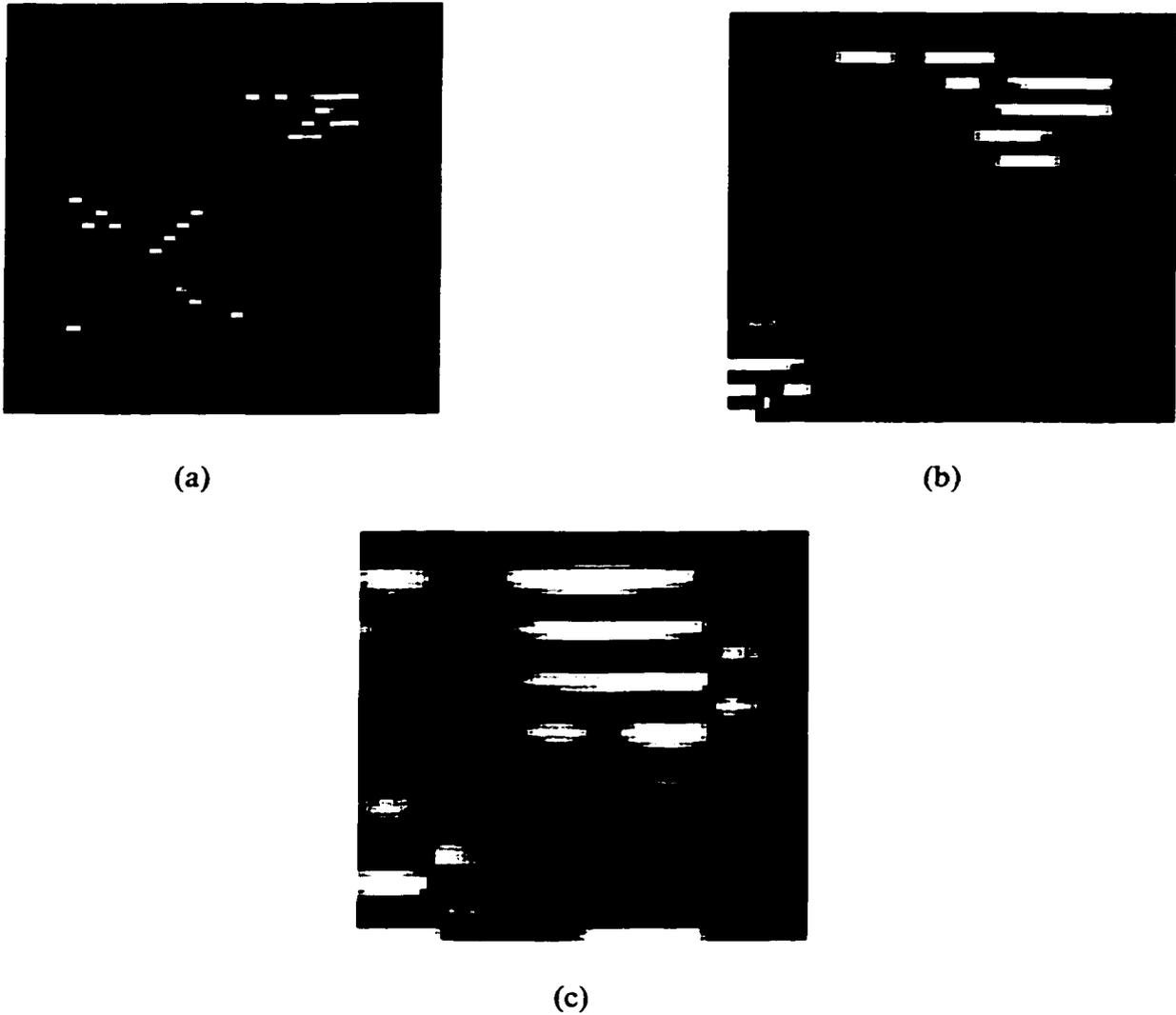


Figure 4.14: Three-level curvature representation from detail coefficients of wavelet decomposition for the top spline face in Figure 4.7. (a): The detailed Feature extracted from the first level. (b): The detailed Feature extracted from the second level. (c): The detailed Feature extracted from the third level.

Table 4.1: Multi-level coding for the top spline face in Figure 4.7

Level	Coefficient Magnitude/ Location in Detail Matrix					Retained Energy
1	1	2	3	4	5	84.9%
	-4.037	3.573	3.339	-2.895	2.441	
	-	<2.00, 0.00>	<2.83, 45.00>	<2.24, 26.57>	<3.00, 90.00>	
	6	7	8	9	10	
	-2.250	2.228	2.220	2.183	1.732	
	<2.24, 63.43>	<5.39, 68.20>	<7.28, 74.05>	<1.00, 90.05>	<4.12, 104.04>	
	11	12	13	14	15	
	-1.637	-1.631	1.608	1.593	-1.528	
	<25.6, 128.66>	<17.00, 151.93>	<13.04, 122.47>	<14.42, 123.69>	<20.62, 112.83>	
16	17	18	19	20		
1.526	1.486	1.483	1.438	1.400		
<17.20, 125.54>	<15.81, 124.70>	<2.24, 116.57>	<1.41, 45.00>	<17.80, 141.84>		
21	22	23	24	25		
-1.351	-1.304	-1.270	1.248	-1.226		
<19.31, 111.25>	<18.79, 115.20>	<20.88, 106.70>	<17.69, 137.29>	<3.66, 108.43>		
2	1	2	3	4	5	91.3%
	-12.119	-10.262	-10.235	-10.071	-8.648	
	-	<14.87, 0.00>	<1.00, 132.27>	<14.14, 357.27>	<16.28, 0.00>	
	6	7	8	9	10	
	8.419	-8.385	6.729	6.707	-6.454	
	<1.41, 357.27>	<16.97, 357.27>	<1.41, 87.27>	<1.00, 42.27>	<15.26, 10.66>	
	11	12	13	14	15	
	6.043	-5.546	5.304	5.058	-4.808	
	<16.28, 354.78>	<17.69, 355.00>	<15.56, 357.27>	<1.00, 222.27>	<14.76, 13.97>	
16	17	18	19	20		
4.742	-4.709	4.696	-4.626	4.503		
<13.45, 354.26>	<14.21, 351.56>	<17.03, 2.03>	<17.03, 352.50>	<14.87, 354.54>		
21	22	23	24	25		
4.465	-4.197	4.099	4.051	-3.879		
<1.41, 177.27>	<2.24, 338.84>	<13.93, 21.23>	<14.42, 8.58>	<15.62, 2.46>		
3	1	2	3	4	5	97.8%
	-22.322	-17.955	-16.340	15.208	-14.838	
	-	<2.00, 0.00>	<7.21, 236.31>	<5.83, 239.04>	<1.00, 0.00>	
	6	7	8	9	10	
	12.776	11.647	11.513	11.304	10.352	
	<6.71, 243.43>	<2.24, 26.57>	<1.00, 90.00>	<1.41, 45.00>	<5.10, 281.31>	
	11	12	13	14	15	
	10.149	6.790	-6.707	6.472	5.589	
	<7.07, 225.00>	<6.40, 231.34>	<1.41, 315.00>	<6.08, 279.46>	<2.24, 333.43>	
16	17	18	19	20		
5.572	4.873	4.532	-3.610	-3.530		
<5.00, 233.13>	<5.66, 225.00>	<4.24, 225.00>	<5.00, 270.00>	<7.81, 230.19>		
21	22	23	24	25		
-3.291	3.282	-3.239	3.222	3.108		
<5.00, 216.87>	<3.16, 288.43>	<6.40, 218.66>	<4.12, 284.04>	<1.41, 225.00>		

4.5.2 Face Contour Coding by Wavelet-Based Zero-Crossings

The main idea of this section is to represent face contours by fine-to-coarse approximations and details based on the wavelet transform zero-crossing representation. In order to build the representation, a radial function describing the face contour is constructed, and its wavelet transform zero-crossing representation is generated.

The following are the steps for the contour coding procedure.

Step I. Transform-Invariant Radial Distance Function

A face contour can be discretized into an ordered list of points and the rectangular coordinates of these points can be used to represent the face contour. However, the coordinate representation is not transform-invariant. Alternatively the face contour can be characterized by a sequence of radial distance between an appropriate reference point $C(u_0, v_0)$ and a point $P(u, v)$ on the face contour. The radial distance function reduces the dimensionality of the problem from a 2D-face contour to a 1D signal and has the transform-invariance property. In this work, the center of the face contour in the parametric space is selected as the reference point. The radial distance function $r(u, v)$ is defined as

$$r(u, v) = |P(u, v) - C(u_0 - v_0)| \quad (4.23)$$

Figure 4.15 illustrates the concept of radial distance function $r(u, v)$. The radial distance function for the contour of the top spline face in Figure 4.7 is shown in Figure 4.16. The contour is sampled by the size of 128.

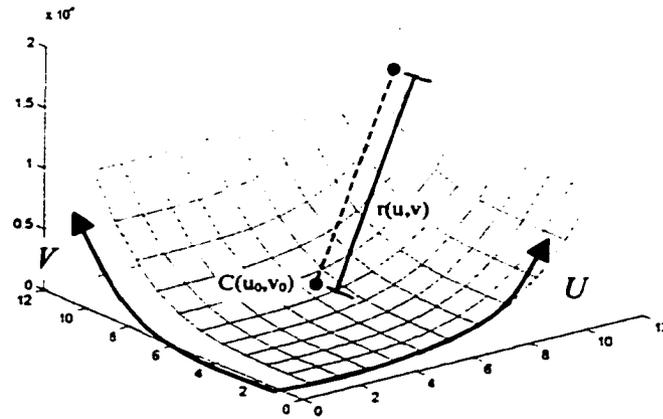


Figure 4.15: Radial distance function for the face contour

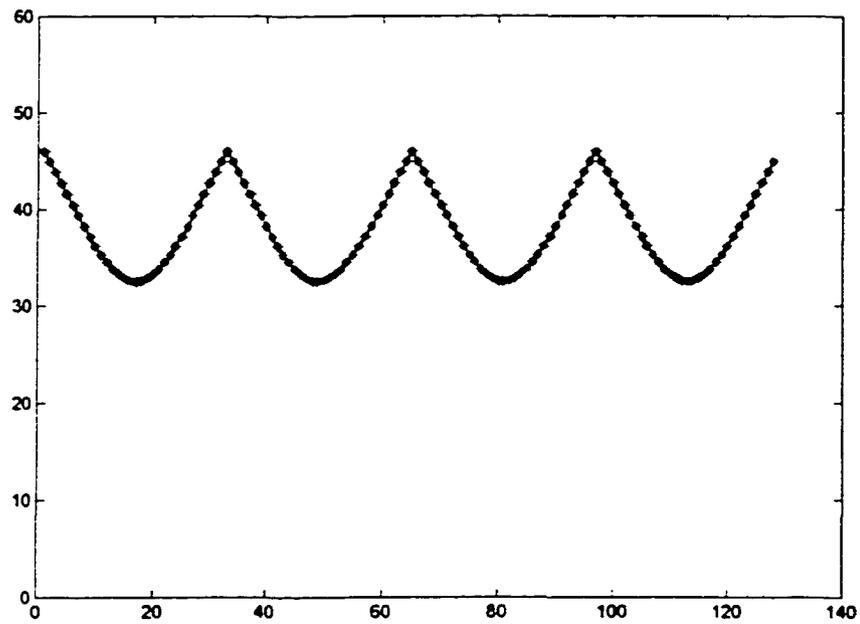


Figure 4.16: Radial distance function for the contour of the top spline face in Figure 4.7

Step II Wavelet-Based Zero-Crossing Representation

The magnitudes of wavelet coefficients of a signal at each resolution level are proportional to the distributed energy of the signal at the associated frequency sub-band [Press et al., 1992]. A few low-resolution levels (high frequencies) can be used for distinguishing objects from one another. It is probable that different objects can have the same representation at a particular resolution level but not all resolution levels. In this work, wavelet approximation and detail coefficients of distance representations at the first three-levels, as shown in Figure 4.17, are used for contour representation and coding.

The zero-crossings of a wavelet transform can provide the locations of multi-scale sharp variation points of the signal. The zero-crossings are formally defined as follows (Hummel and Moniot, 1992):

The zero crossings of function $f(u,v)$ refer to the point set

$\partial\{(u,v) | f(u,v) < 0\} \cap \partial\{(u,v) | f(u,v) > 0\}$, where ∂S refers to the boundary of the set S .

In the zero-crossing representation, values of wavelet coefficients between two zero-crossings are replaced by these coefficients' average value. The Resulting three-level wavelet zero-crossing representations for the contour of the top spline face in Figure 4.7 are listed in Table 4.2. For level 1, the first row in the table (1, 2, 3~4, 5, 6, 7~28, 29~30, 31~32, 33~34, and 35~36) represent the first thirty-six sampling points of the spline face contour. The second row is the zero-crossing data for the corresponding sampling points. The same is with level 2 and level 3. The results are also plotted in Figures 4.18, 4.19, and 4.20 respectively.

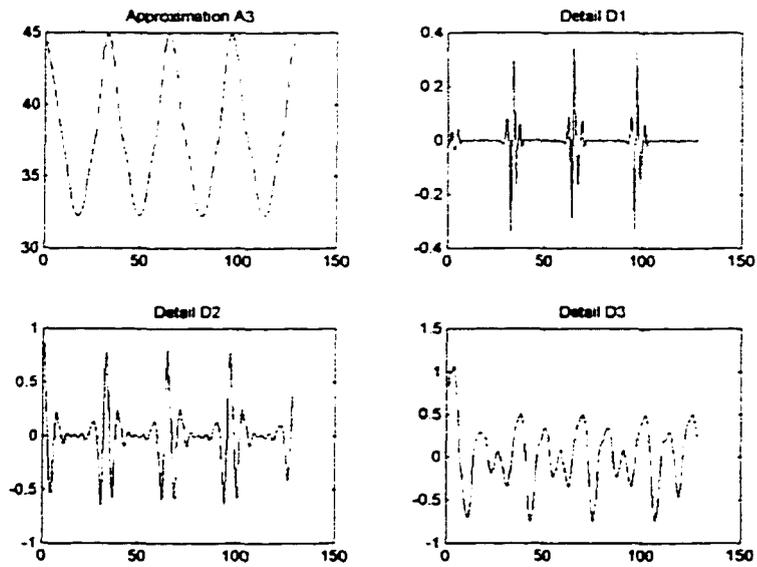


Figure 4.17: Three-level wavelet decomposition of radial distance function in Figure 4.16

Table 4.2: Multi-level coding for the spline face contour in Figure 4.7

Level	Wavelet Zero-Crossing Representation									
1	1	2	3~4	5	6	7~28	29~30	31~32	33~34	35~36
	-0.016	0.048	-0.031	0.043	-0.017	0	0.058	-0.182	0.199	-0.090
	37	38	39~60	61~62	63~64	65~66	67~68	69	70	71~92
	0.076	-0.027	0	0.058	-0.182	0.199	-0.090	0.076	-0.027	0
	93~94	95~96	97~98	99~100	101	102	103~128			
0.058	-0.182	0.199	-0.090	0.076	-0.027	0				
2	1~2	3~5	6~9	10~11	12~21	22~23	24~27	28~31	32~34	35~37
	0.566	-0.381	0.129	-0.066	0	-0.035	0.099	-0.330	0.578	-0.394
	38~41	42~43	44~53	54~55	56~59	60~63	64~66	67~69	70~73	74~75
	0.141	-0.079	0	-0.035	0.099	-0.330	0.578	-0.394	0.141	-0.079
	76~85	86~87	88~91	92~95	96~98	99~101	102~105	106~107	108~117	118~119
	0	-0.035	0.099	-0.330	0.578	-0.394	0.141	-0.079	0	-0.033
	120~123	124~127	128							
0.0736	-0.237	0.371								
3	1	7	15	22	26	28	34	41	47	54
	0.835	-0.390	0.232	-0.134	0.068	-0.187	0.366	-0.437	0.239	-0.162
	58	60	66	73	79	86	90	92	98	105
	0.077	-0.186	0.366	-0.437	0.239	-0.162	0.077	-0.186	0.366	-0.415
	111	117	122							
0.221	-0.282	0.432								

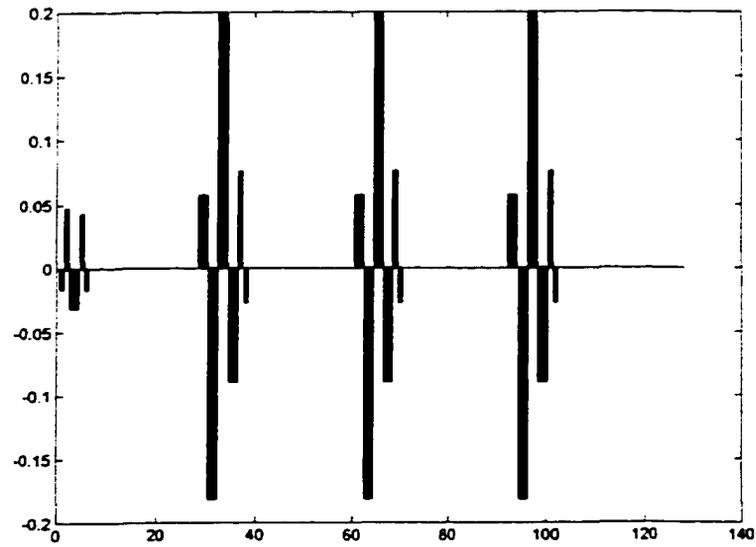


Figure 4.18: Level-1 zero-crossing representation of spline face contour in Figure 4.7

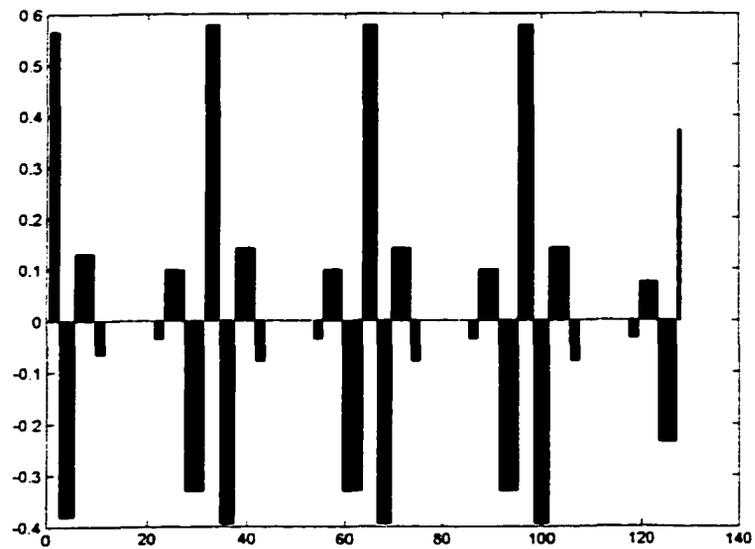


Figure 4.19: Level-2 zero-crossing representation of spline face contour in Figure 4.7

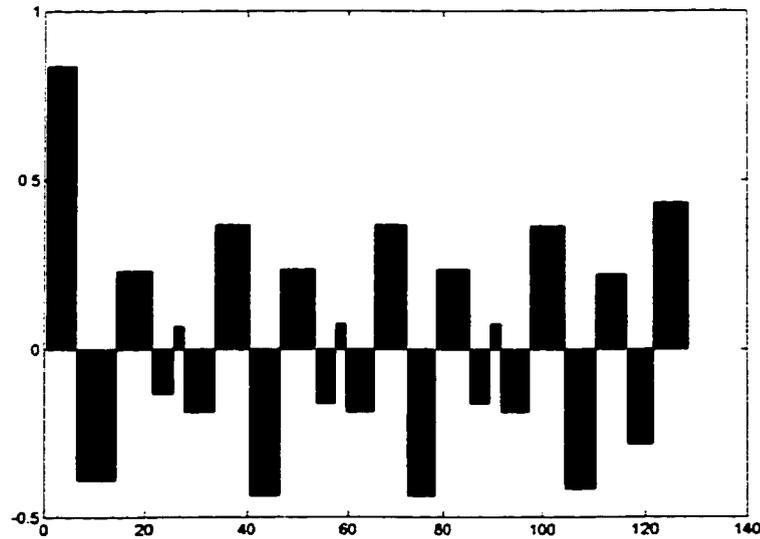


Figure 4.20: Level-3 zero-crossing representation of spline face contour in Figure 4.7

4.6 An Example

In this section, the ACIS model in Figure 4.21, which has been ranked by the algorithms presented in Chapter 3, will be used for demonstrating wavelet-based object and form feature coding. The model has eleven faces in total, two of which are free form faces, three are conical faces, and the rest are flat faces.

The object coding procedure starts with constructing curvature-based representation for each face region. The maximum principal curvature map for free form face-5 is given in Figure 4.22. Its three-level wavelet decompositions are shown in Figures 4.23, 4.24, and 4.25 respectively. Afterward three-level curvature-based coding for the face region is listed in Table 4.3 and visualized in Figure 4.26. The maximum principal curvature map for free

form face-8 is given in Figure 4.27. Its three-level wavelet decompositions are shown in Figures 4.28, 4.29, and 4.30 respectively. Three-level curvature-based coding for its face region is listed in Table 4.4 and presented in Figure 4.31. The maximum principal curvatures for the two conical surfaces and one cylindrical surface carry the same value over all face regions. Therefore, their wavelet approximations at the same level are uniformly distributed and their wavelet details at three-levels are all zeros. What distinguishes the three conical surfaces is that their wavelet approximations at the same level have different values based on the radius of each conical surface. For the other flat faces, both their wavelet approximations and details are zeros.

The next step of the object coding procedure is to code the boundary of each face in the object. For face-1 as an example, the transform-invariant radial distance function for its contour is plotted in Figure 4.32. Its three-level wavelet decompositions are then shown in Figure 4.33. Finally three-level wavelet zero-crossing representations for the face contour are listed in Table 4.5 and visualized in Figure 4.34. The same process is applied to the other ten faces and the results are listed in Tables 4.6 through 4.12 and displayed in Figures 4.35 through 4.55. Since face-3 and face-4 have similar contours, only the results of face-3 are presented. It is the same with face-9 and face-10. So only the results of face-9 are displayed. Face-7, the bottom face of the cylinder hole, is not presented, as the radial distance function for its contour is constant and all its three-level zero-crossing representations are zeros.

Lastly a face representation is defined as the combination of its region codes and contour codes for all the faces in the object. The object is then represented by a face adjacency graph (FAG). FAG was originally developed by Joshi and Chang (1988) to represent the boundary of an object. In the graph each face on the object is represented by a node and each edge is

represented by an arc. An arc is assigned an attribute value +1 for a convex edge and -1 for a concave edge. FAG can be further divided into sub-graphs based on the clustering information obtained from Chapter 3, Section 3.2, Stage 5. The resulting object representation is presented in Figure 4.56.

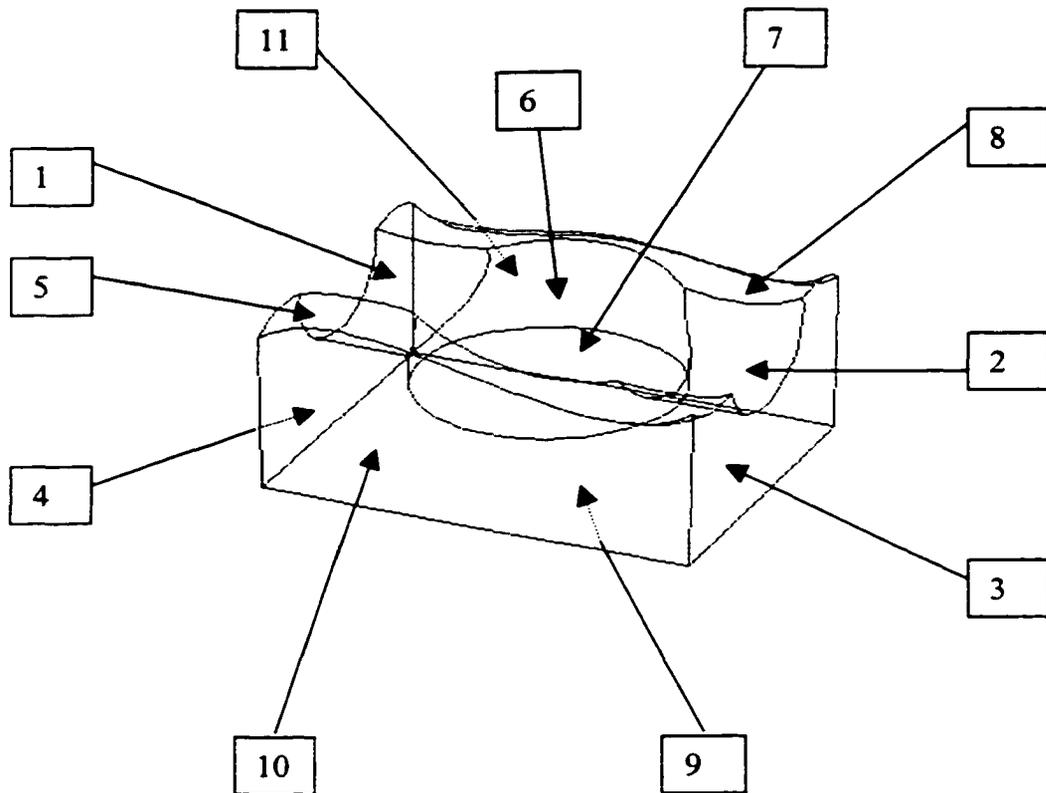


Figure 4.21: An ACIS part with free-form features on its top surfaces

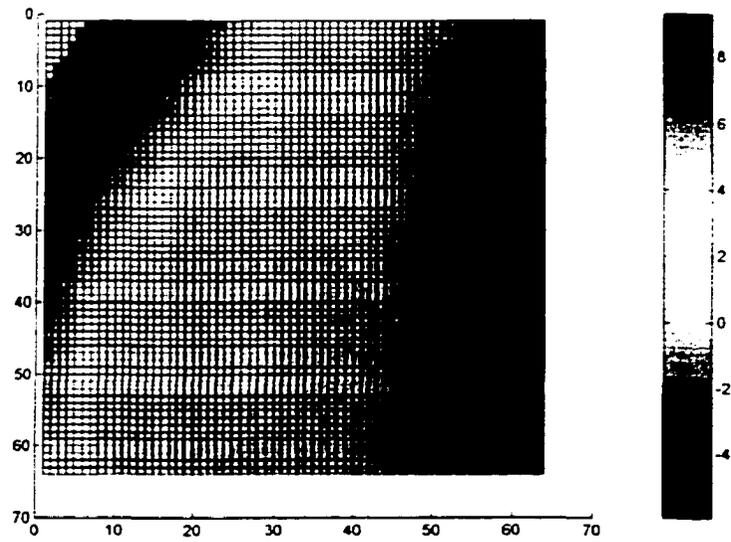


Figure 4.22: Map of maximum principal curvature of spline surface-5 in Figure 4.21

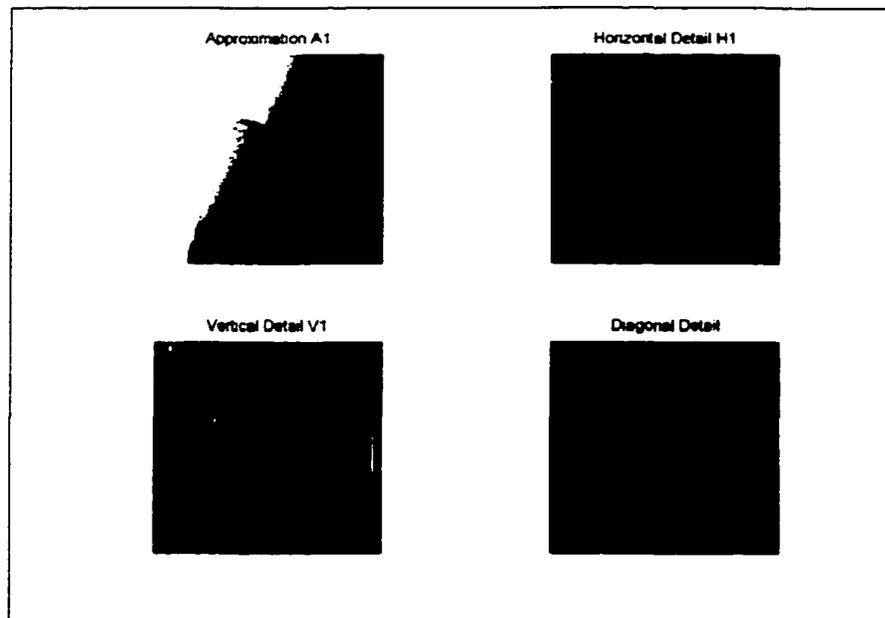


Figure 4.23: The level-1 decomposition of curvature map in Figure 4.22

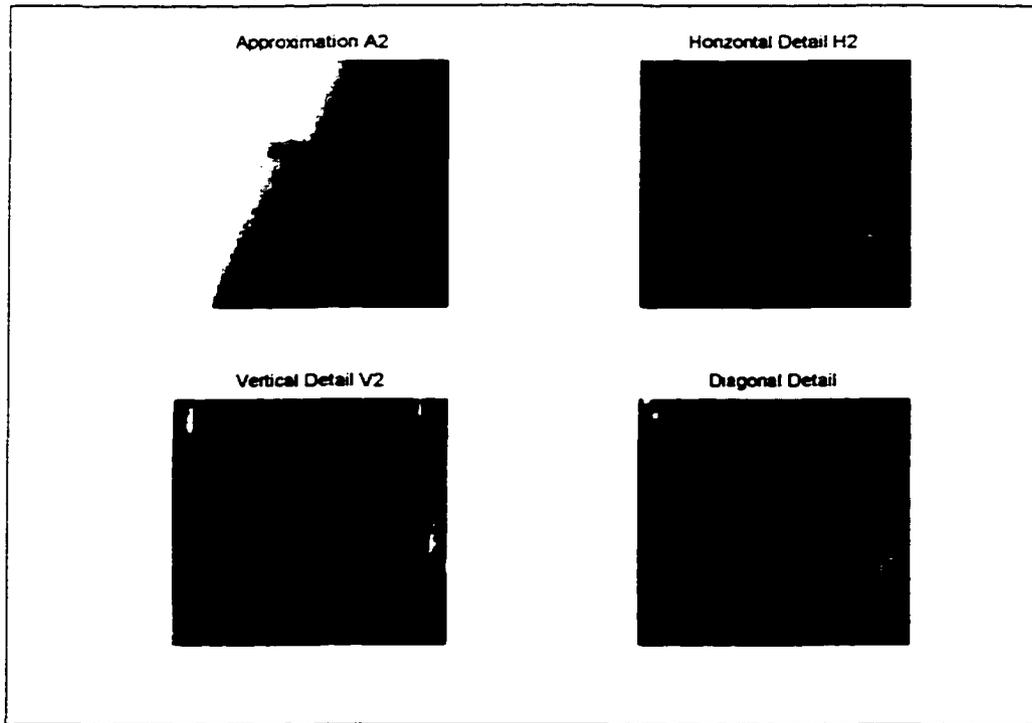


Figure 4.24: The level-2 decomposition of curvature map in Figure 4.22

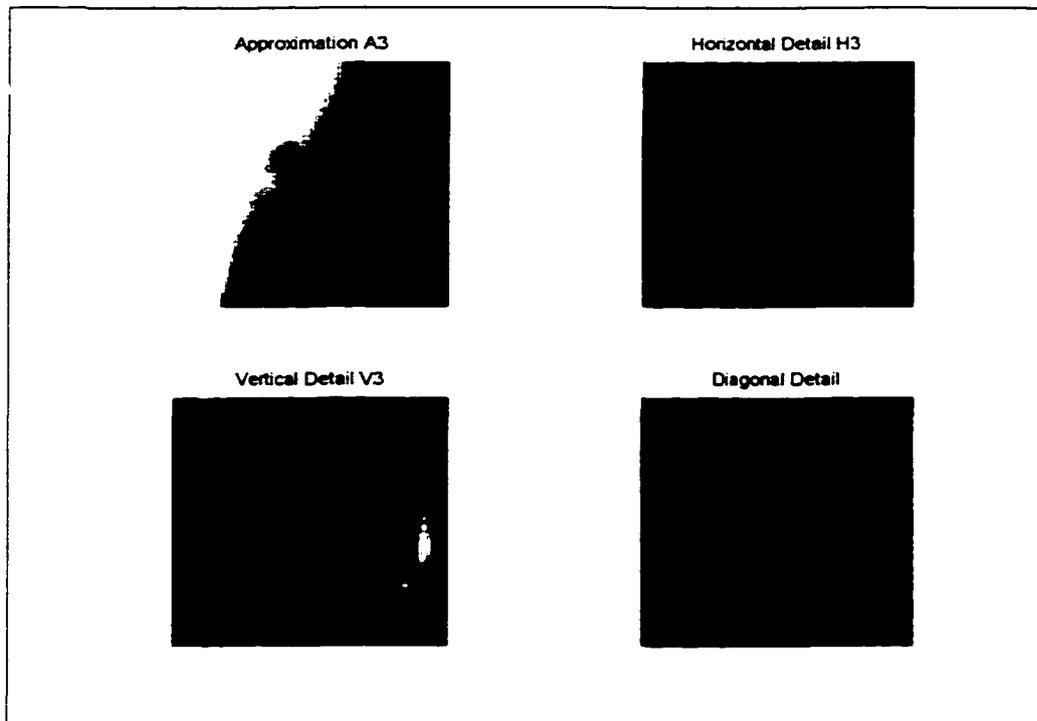
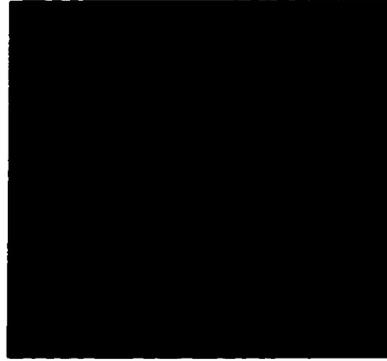


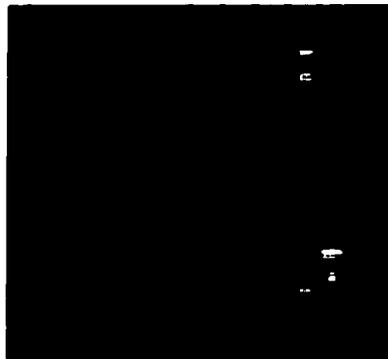
Figure 4.25: The level-3 decomposition of curvature map in Figure 4.22

Table 4.3: Multi-level coding for the spline surface-5 in Figure 4.21

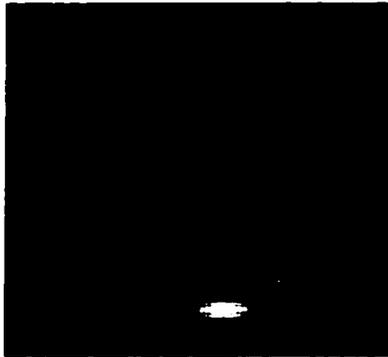
Level	Coefficient Magnitude/ Location in Detail Matrix					Retained Energy
1	1	2	3	4	5	89.0%
	-0.887	0.639	-0.637	0.613	-0.592	
	-	<6.40, 0.00>	<2.23, 192.09>	<4.47, 12.09>	<5.83, 7.70>	
	6	7	8	9	10	
	0.589	0.559	0.491	-0.486	-0.468	
	<3.16, 20.22>	<5.00, 345.53>	<1.41, 173.66>	<7.81, 348.47>	<4.24, 353.66>	
	11	12	13	14	15	
	0.440	0.419	-0.416	0.405	0.397	
	<4.12, 24.62>	<3.00, 338.66>	<2.83, 353.66>	<7.21, 342.35>	<1.00, 308.66>	
	16	17	18	19	20	
0.372	-0.370	-0.360	0.358	-0.334		
<1.00, 38.66>	<1.41, 353.66>	<5.66, 353.66>	<6.40, 347.32>	<3.61, 4.97>		
21	22	23	24	25		
0.324	-0.314	-0.304	-0.301	0.291		
<11.18, 282.09>	<5.39, 196.86>	<7.07, 353.66>	<8.94, 335.22>	<3.61, 184.97>		
2	1	2	3	4	5	90.0%
	-2.322	-2.189	-1.932	-1.771	-1.646	
	-	<9.06, 0.00>	<10.05, 359.37>	<1.00, 353.66>	<1.41, 38.66>	
	6	7	8	9	10	
	-1.416	-1.331	1.301	-1.268	1.223	
	<2.00, 353.66>	<9.90, 38.66>	<1.00, 83.66>	<16.64, 50.92>	<1.41, 128.66>	
	11	12	13	14	15	
	-1.167	-1.080	0.923	-0.923	-0.890	
	<16.12, 53.91>	<8.06, 0.78>	<2.00, 173.66>	<2.24, 147.09>	<17.20, 48.12>	
	16	17	18	19	20	
0.822	-0.807	0.787	-0.784	-0.770		
<13.04, 51.19>	<11.70, 63.68>	<12.37, 69.62>	<10.63, 42.47>	<13.60, 100.76>		
21	22	23	24	25		
0.713	0.703	0.697	-0.688	-0.682		
<22.24, 20.22>	<14.56, 99.61>	<3.16, 12.09>	<13.34, 96.95>	<12.17, 74.20>		
3	1	2	3	4	5	94.5%
	4.585	-2.458	-2.171	2.134	-1.864	
	-	<7.07, 0.00>	<7.81, 354.81>	<7.81, 5.19>	<6.32, 26.57>	
	6	7	8	9	10	
	-1.843	-1.809	1.727	-1.718	1.631	
	<6.40, 6.34>	<6.08, 35.54>	<5.10, 303.69>	<6.00, 45.00>	<6.08, 305.54>	
	11	12	13	14	15	
	1.620	1.440	1.363	-1.347	1.334	
	<8.49, 0.00>	<1.41, 180.00>	<4.12, 310.96>	<6.08, 54.46>	<1.00, 315.00>	
	16	17	18	19	20	
1.328	1.276	1.245	-1.045	1.012		
<2.00, 315.00>	<5.10, 56.31>	<3.16, 296.57>	<6.32, 333.43>	<2.34, 288.43>		
21	22	23	24	25		
0.997	0.974	0.954	-0.906	0.872		
<7.21, 11.31>	<5.00, 45.00>	<4.47, 341.57>	<1.00, 225.00>	<4.47, 18.43>		



(a)



(b)



(c)

Figure 4.26: Three-level curvature representation from detail coefficients of wavelet decomposition for the spline face-5 in Figure 4.21. (a): Level-1 details of wavelet decomposition. (b): Level-2 details of wavelet decomposition. (c) Level-3 details of wavelet decomposition.

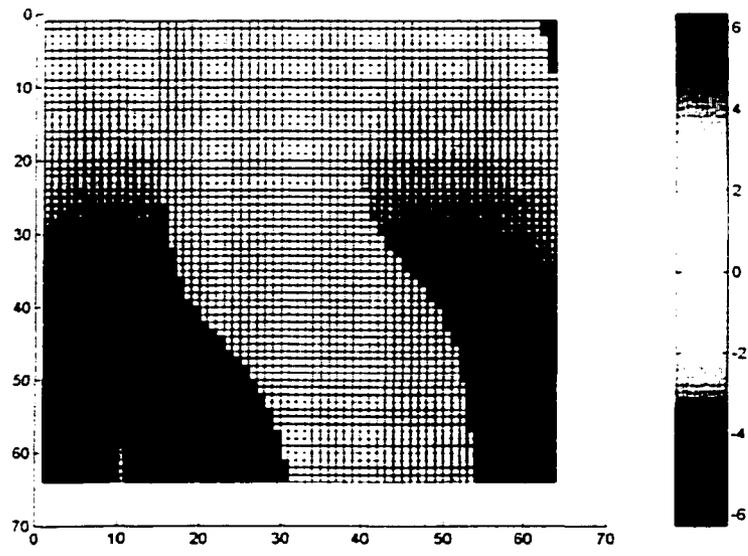


Figure 4.27: Map of principal curvature of spline surface-8 in Figure 4.21

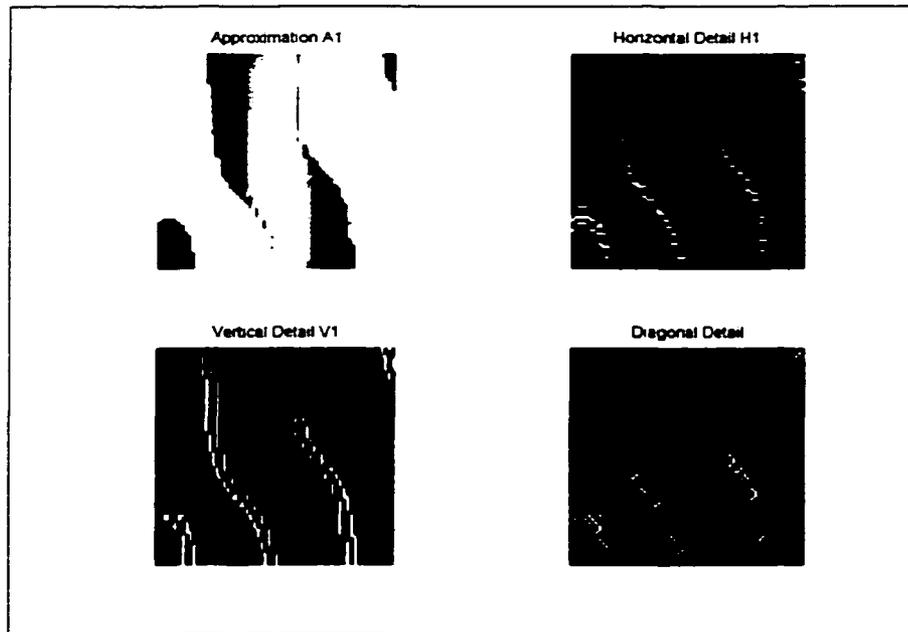


Figure 4.28: The level-1 decomposition of curvature map in Figure 4.27

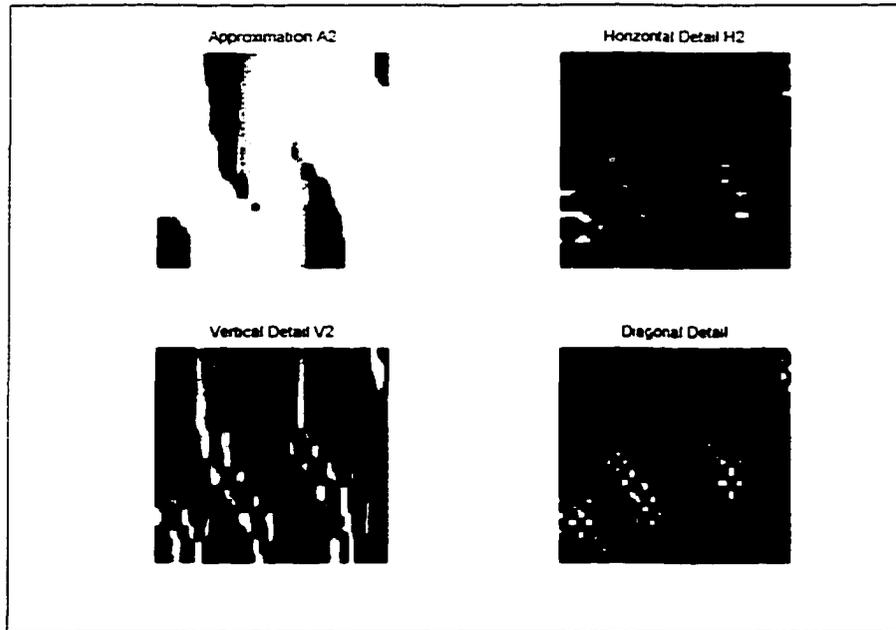


Figure 4.29: The level-2 decomposition of curvature map in Figure 4.27

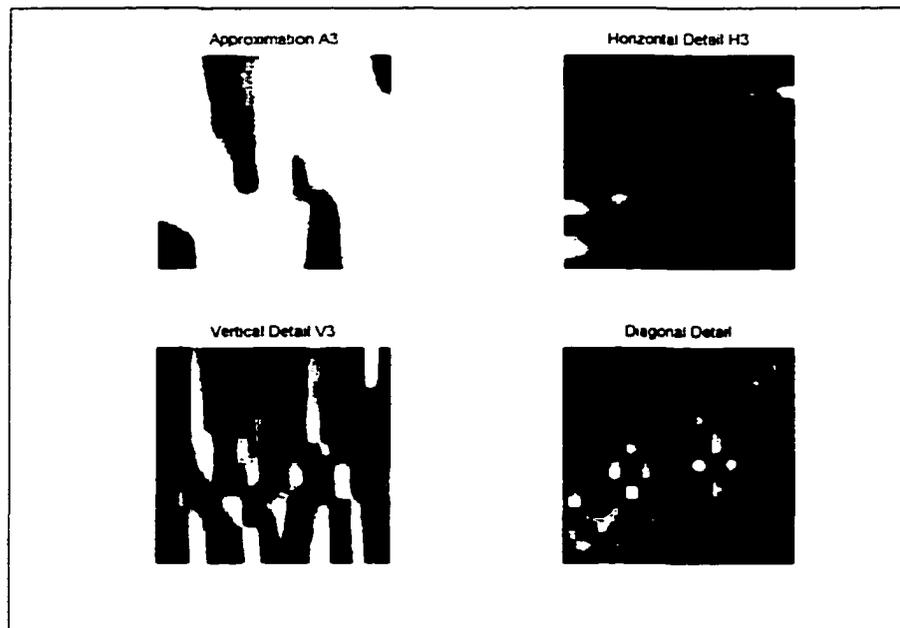
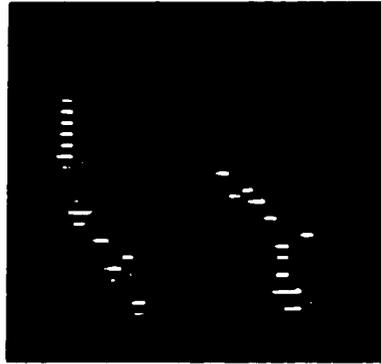


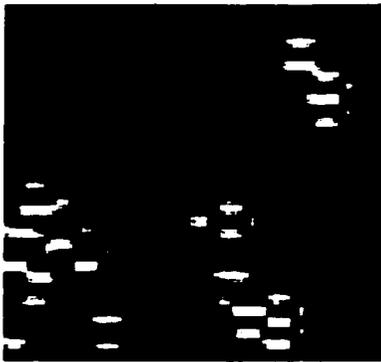
Figure 4.30: The level-3 decomposition of curvature map in Figure 4.27

Table 4.4: Multi-level coding for the spline surface-8 in Figure 4.21

Level	Coefficient Magnitude/ Location in Detail Matrix					Retained Energy
1	1	2	3	4	5	80.0%
	-4.523	4.497	-3.758	-3.674	-3.519	
	-	<19.70, 0.00>	<5.10, 125.27>	<9.43, 325.97>	<5.83, 324.92>	
	6	7	8	9	10	
	-3.409	2.950	-2.930	-2.844	2.844	
	<17.26, 13.95>	<16.00, 23.96>	<18.38, 1.58>	<7.07, 122.09>	<2.83, 338.96>	
	11	12	13	14	15	
	-2.811	-2.729	2.709	-2.658	-2.617	
	<6.08, 123.42>	<1.00, 23.96>	<12.65, 42.40>	<1.00, 113.96>	<8.06, 121.08>	
16	17	18	19	20		
2.613	-2.596	-2.579	-2.549	2.501		
<13.15, 32.71>	<5.66, 338.96>	<14.14, 32.09>	<15.03, 27.77>	<17.72, 7.57>		
21	22	23	24	25		
-2.500	-2.487	-2.433	-2.390	-2.249		
<18.03, 4.52>	<19.11, 17.95>	<9.06, 120.30>	<19.31, 2.71>	<10.05, 119.67>		
2	1	2	3	4	5	86.0%
	10.661	10.464	-8.843	7.891	6.625	
	-	<10.20, 0.00>	<15.26, 42.92>	<1.00, 191.31>	<3.16, 262.87>	
	6	7	8	9	10	
	-5.852	-5.724	-5.614	-5.456	4.831	
	<2.24, 74.74>	<14.76, 39.61>	<3.16, 82.88>	<1.41, 326.31>	<15.00, 48.18>	
	11	12	13	14	15	
	4.805	-4.770	-4.538	-4.500	4.284	
	<15.62, 51.16>	<9.22, 23.83>	<9.06, 4.97>	<4.00, 281.31>	<3.00, 11.31>	
16	17	18	19	20		
-4.268	-3.964	-3.898	-3.866	-3.813		
<5.00, 334.44>	<11.18, 1.00>	<2.00, 11.31>	<11.40, 356.05>	<11.70, 351.32>		
21	22	23	24	25		
3.790	-3.529	3.515	3.514	3.426		
<2.24, 37.88>	<2.83, 56.31>	<8.25, 25.35>	<1.41, 236.31>	<10.44, 354.61>		
3	1	2	3	4	5	98.5%
	-27.429	-21.808	15.155	12.546	12.354	
	-	<1.00, 0.00>	<1.00, 90.00>	<4.12, 165.96>	<1.41, 45.00>	
	6	7	8	9	10	
	11.408	10.649	-9.484	8.682	7.766	
	<6.40, 218.66>	<7.07, 225.00>	<2.24, 116.57>	<5.83, 210.96>	<3.16, 198.43>	
	11	12	13	14	15	
	6.510	6.162	-5.734	5.393	5.180	
	<1.41, 135.00>	<1.00, 270.00>	<1.00, 180.00>	<1.41, 315.00>	<2.83, 135.00>	
16	17	18	19	20		
-5.122	-4.924	-4.700	4.166	-3.950		
<5.39, 158.20>	<3.61, 213.69>	<5.10, 168.69>	<2.24, 296.57>	<4.00, 180.00>		
21	22	23	24	25		
3.613	-3.462	3.273	3.038	-2.773		
<3.16, 288.44>	<2.00, 90.00>	<5.10, 281.31>	<4.12, 284.04>	<1.41, 225.00>		



(a)



(b)



(c)

Figure 4.31: Three-level curvature representation from detail coefficients of wavelet decomposition for the spline face-8 in Figure 4.21. (a): Level-1 details of wavelet decomposition. (b): Level C: Level-3 details of wavelet decomposition. (c): Level-3 details of wavelet decomposition.

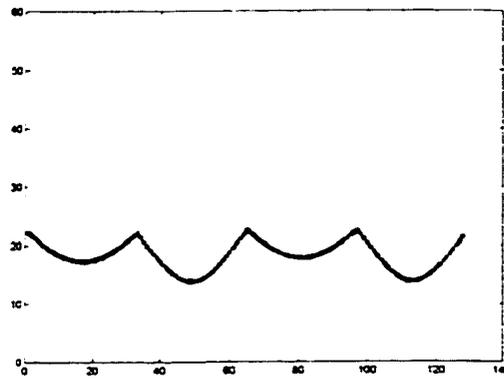


Figure 4.32: Radial distance function for the contour of the conical face-1 in Figure 4.21

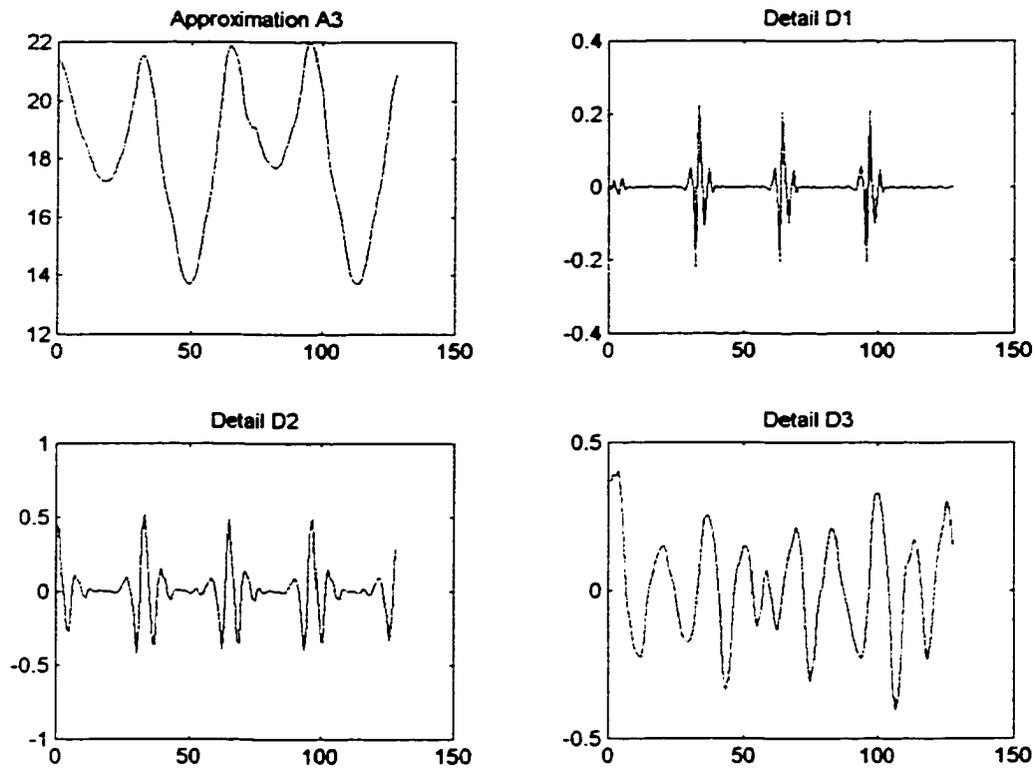
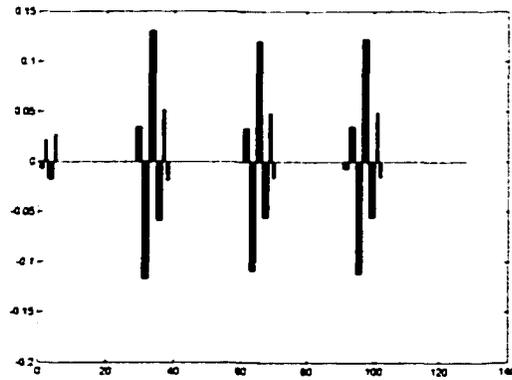


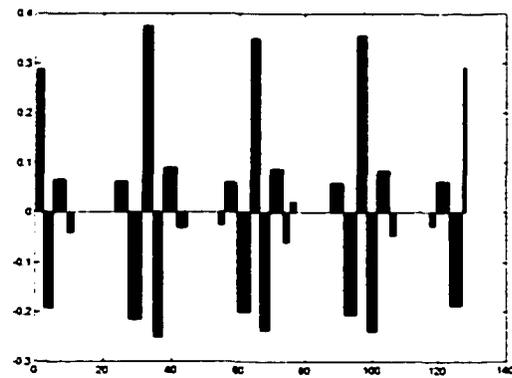
Figure 4.33: Three-level wavelet decomposition of radial distance function in Figure 4.32

Table 4.5: Multi-level coding for the contour of conical face-1 in Figure 4.21

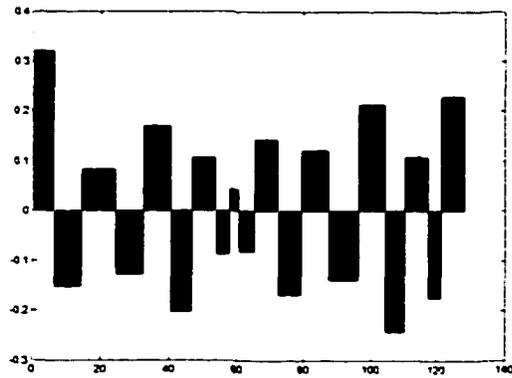
Level	Wavelet Zero-Crossing Representation									
1	<i>1</i>	<i>2</i>	<i>3-4</i>	<i>5</i>	<i>6-28</i>	<i>29-30</i>	<i>31-32</i>	<i>33-34</i>	<i>35-36</i>	<i>37</i>
	-0.007	0.022	-0.017	0.028	0	0.035	-0.117	0.130	-0.059	0.052
	<i>38</i>	<i>39-60</i>	<i>61-62</i>	<i>63-64</i>	<i>65-66</i>	<i>67-68</i>	<i>69</i>	<i>70</i>	<i>71-90</i>	<i>91-92</i>
	-0.019	0	0.034	-0.109	0.121	-0.055	0.049	-0.016	0	-0.007
	<i>93-94</i>	<i>95-96</i>	<i>97-98</i>	<i>99-100</i>	<i>101</i>	<i>102</i>	<i>103-128</i>			
0.035	-0.112	0.122	-0.056	0.050	-0.016	0				
2	<i>1-2</i>	<i>3-5</i>	<i>6-9</i>	<i>10-11</i>	<i>12-23</i>	<i>24-27</i>	<i>28-31</i>	<i>32-34</i>	<i>35-37</i>	<i>38-41</i>
	0.289	-0.192	0.067	-0.040	0	0.062	-0.215	0.375	-0.252	0.090
	<i>42-44</i>	<i>45-53</i>	<i>54-55</i>	<i>56-59</i>	<i>60-63</i>	<i>64-66</i>	<i>67-69</i>	<i>70-73</i>	<i>74-75</i>	<i>76-77</i>
	-0.031	0	-0.024	0.061	-0.200	0.350	-0.237	0.088	-0.059	0.021
	<i>78-87</i>	<i>88-91</i>	<i>92-95</i>	<i>96-98</i>	<i>99-101</i>	<i>102-105</i>	<i>106-107</i>	<i>108-117</i>	<i>118-119</i>	<i>120-123</i>
	0	0.060	-0.206	0.356	-0.238	0.084	-0.046	0	-0.027	0.062
	<i>124-127</i>	<i>128</i>								
-0.187	0.293									
3	<i>1-6</i>	<i>7-14</i>	<i>15-24</i>	<i>25-32</i>	<i>33-40</i>	<i>41-46</i>	<i>47-53</i>	<i>54-57</i>	<i>58-60</i>	<i>61-65</i>
	0.321	-0.152	0.083	-0.128	0.170	-0.203	0.106	-0.086	0.043	-0.081
	<i>66-72</i>	<i>73-79</i>	<i>80-87</i>	<i>88-96</i>	<i>97-104</i>	<i>105-110</i>	<i>111-117</i>	<i>118-121</i>	<i>122-128</i>	
	0.144	-0.168	0.122	-0.139	0.213	-0.243	0.109	-0.174	0.229	



(a)



(b)



(c)

Figure 4.34: Three-level wavelet zero-crossing representation of the contour of conical face-1 in Figure 4.21. (a): Level-1 wavelet zero-crossing representation. (b): C: Level-2 wavelet zero-crossing representation. (c): Level-3 wavelet zero-crossing representation.

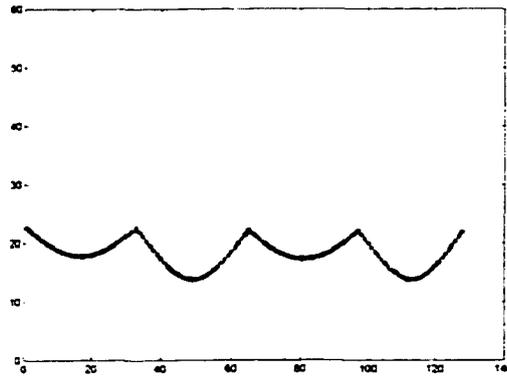


Figure 4.35: Radial distance function for the contour of conical face-2 in Figure 4.21

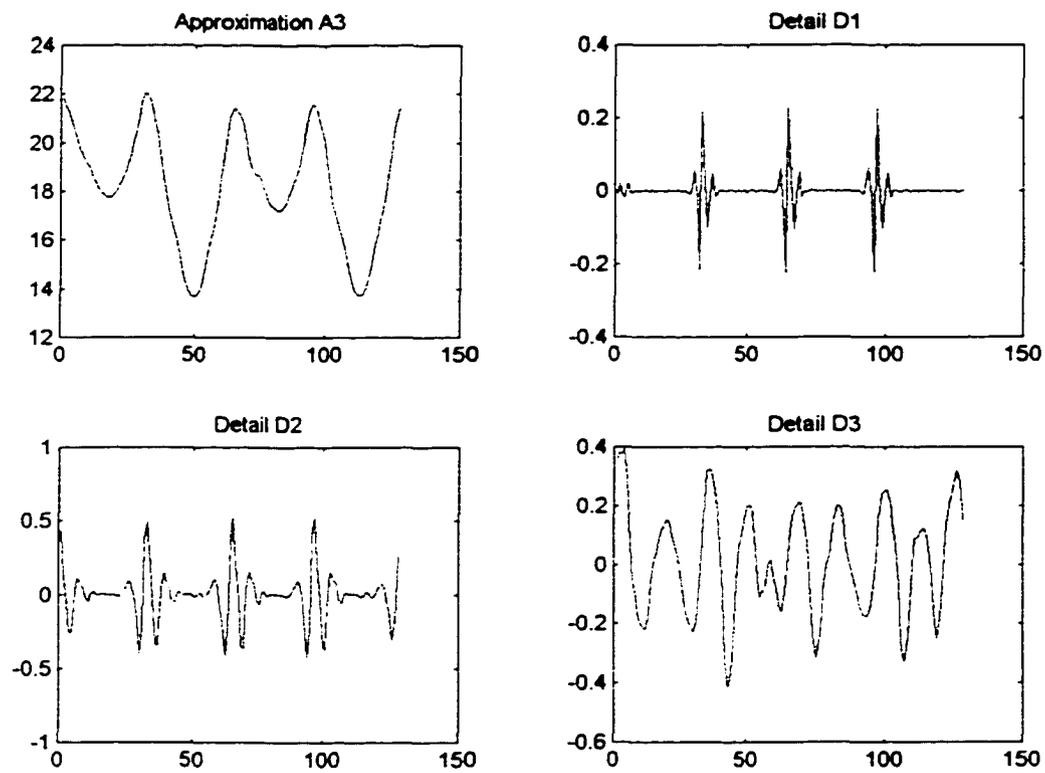
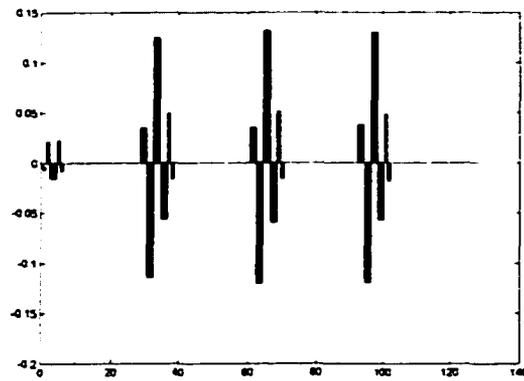


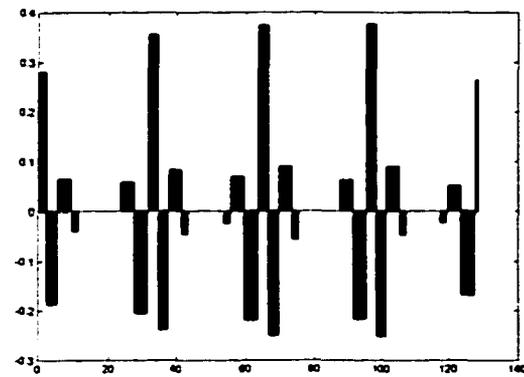
Figure 4.36: Three-level wavelet decomposition of radial distance function in Figure 4.35

Table 4.6: Multi-level coding for the contour of conical face-2 in Figure 4.21

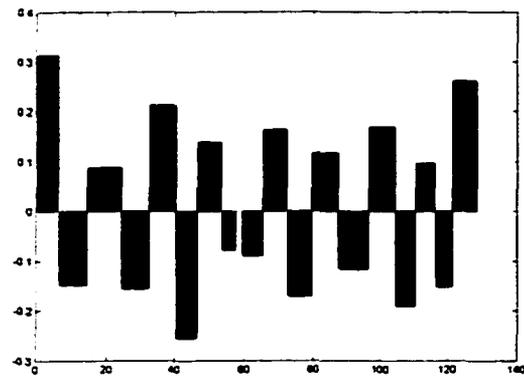
Level	Wavelet Zero-Crossing Representation									
1	<i>1</i>	<i>2</i>	<i>3-4</i>	<i>5</i>	<i>6</i>	<i>7-28</i>	<i>29-30</i>	<i>31-32</i>	<i>33-34</i>	<i>35-36</i>
	-0.007	0.021	-0.015	0.023	-0.009	0	0.035	-0.114	0.125	-0.056
	<i>37</i>	<i>38</i>	<i>39-60</i>	<i>61-62</i>	<i>63-64</i>	<i>65-66</i>	<i>67-68</i>	<i>69</i>	<i>70</i>	<i>71-92</i>
	0.050	-0.016	0	0.0353	-0.120	0.132	-0.059	0.051	-0.016	0
	<i>93-94</i>	<i>95-96</i>	<i>97-98</i>	<i>99-100</i>	<i>101</i>	<i>102</i>	<i>103-128</i>			
0.038	-0.120	0.130	-0.057	0.049	-0.019	0				
2	<i>1-2</i>	<i>3-5</i>	<i>6-9</i>	<i>10-11</i>	<i>12-23</i>	<i>24-27</i>	<i>28-31</i>	<i>32-34</i>	<i>35-37</i>	<i>38-41</i>
	0.281	-0.187	0.065	-0.040	0	0.059	-0.205	0.356	-0.238	0.084
	<i>42-43</i>	<i>44-53</i>	<i>54-55</i>	<i>56-59</i>	<i>60-63</i>	<i>64-66</i>	<i>67-69</i>	<i>70-73</i>	<i>74-75</i>	<i>76-87</i>
	-0.046	0	-0.024	0.070	-0.219	0.374	-0.250	0.090	-0.056	0
	<i>88-91</i>	<i>92-95</i>	<i>96-98</i>	<i>99-101</i>	<i>102-105</i>	<i>106-107</i>	<i>108-117</i>	<i>118-119</i>	<i>120-123</i>	<i>124-127</i>
	0.063	-0.217	0.377	-0.253	0.091	-0.048	0	-0.022	0.052	-0.168
	<i>128</i>									
0.265										
3	<i>1</i>	<i>7</i>	<i>15</i>	<i>25</i>	<i>33</i>	<i>41</i>	<i>47</i>	<i>54</i>	<i>58</i>	<i>60</i>
	0.313	-0.148	0.089	-0.156	0.213	-0.255	0.139	-0.78	0	-0.900
	<i>66</i>	<i>73</i>	<i>80</i>	<i>88</i>	<i>97</i>	<i>105</i>	<i>111</i>	<i>117</i>	<i>122</i>	
0.164	-0.171	0.117	-0.116	0.170	-0.191	0.097	-0.152	0.262		



(a)



(b)



(c)

Figure 4.37: Three-level wavelet zero-crossing representation of the contour of conical face-2 in Figure 4.21. (a): Level-1 wavelet zero-crossing representation. (b): C: Level-2 wavelet zero-crossing representation. (c): Level-3 wavelet zero-crossing representation.

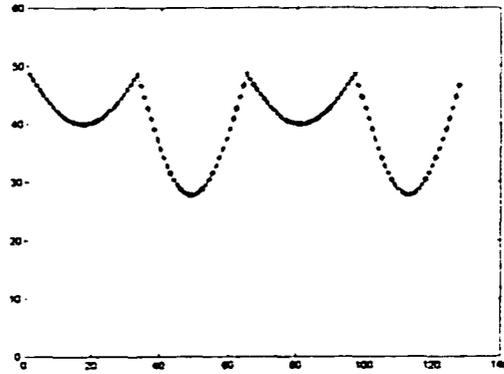


Figure 4.38: Radial distance function for the contour of face-3 in Figure 4.21

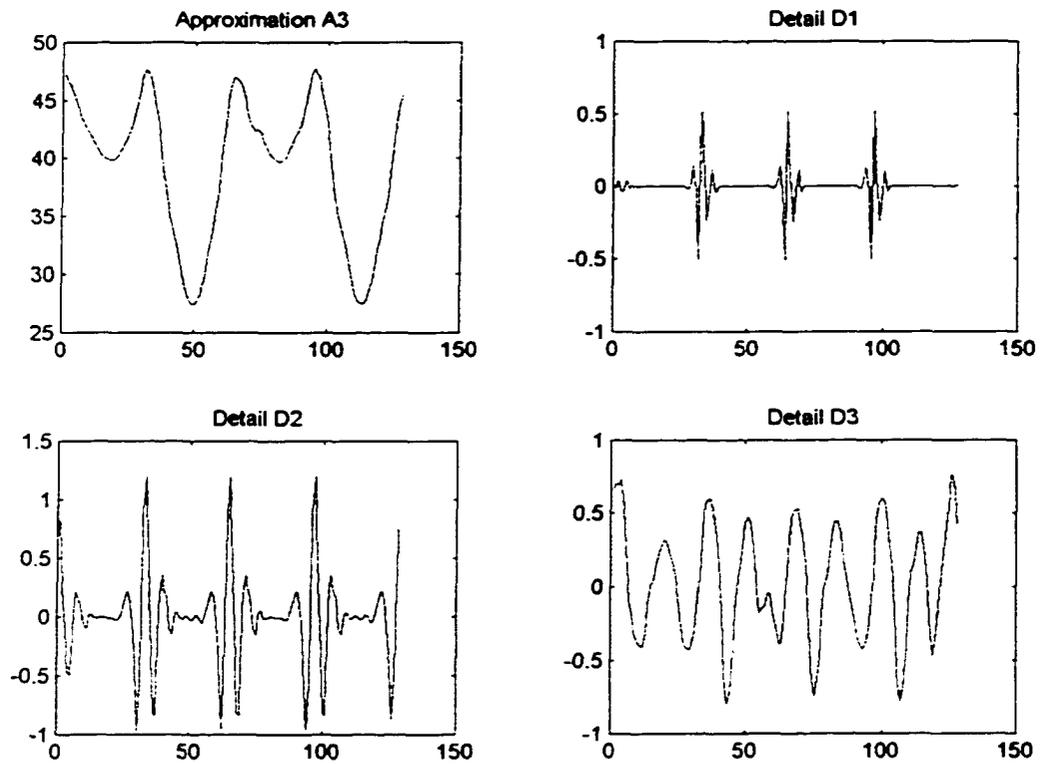
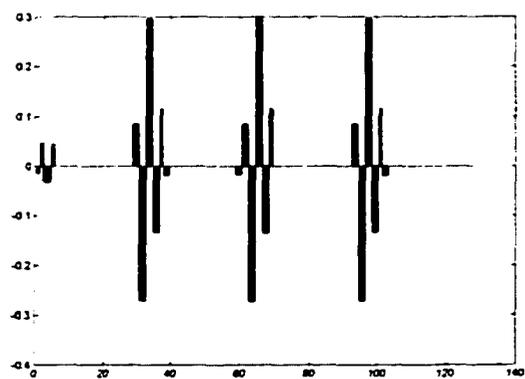


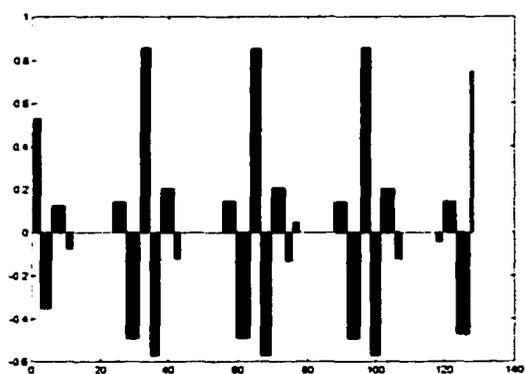
Figure 4.39: Three-level wavelet decomposition of radial distance function in Figure 4.38

Table 4.7: Multi-level coding for the contour of face-3 in Figure 4.21

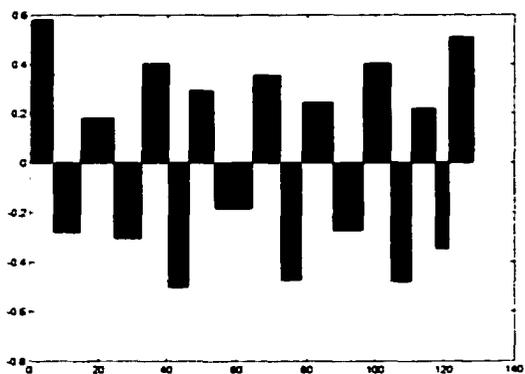
Level	Wavelet Zero-Crossing Representation									
1	<i>1</i>	<i>2</i>	<i>3-4</i>	<i>5</i>	<i>6-28</i>	<i>29-30</i>	<i>31-32</i>	<i>33-34</i>	<i>35-36</i>	<i>37</i>
	-0.015	0.047	-0.032	0.045	0	0.085	-0.272	0.296	-0.133	0.116
	<i>38-39</i>	<i>40-58</i>	<i>59-60</i>	<i>61-62</i>	<i>63-64</i>	<i>65-66</i>	<i>67-68</i>	<i>69</i>	<i>70-92</i>	<i>93-94</i>
	-0.019	0	-0.017	0.084	-0.273	0.300	-0.135	0.116	0	0.085
	<i>95-96</i>	<i>97-98</i>	<i>99-100</i>	<i>101</i>	<i>102-103</i>	<i>104-128</i>				
-0.272	0.296	-0.133	0.116	-0.019	0					
2	<i>1-2</i>	<i>3-5</i>	<i>6-9</i>	<i>10-11</i>	<i>12-23</i>	<i>24-27</i>	<i>28-31</i>	<i>32-34</i>	<i>35-37</i>	<i>38-41</i>
	0.530	-0.352	0.123	-0.078	0	0.141	-0.493	0.857	-0.573	0.205
	<i>42-43</i>	<i>44-55</i>	<i>56-59</i>	<i>60-63</i>	<i>64-66</i>	<i>67-69</i>	<i>70-73</i>	<i>74-75</i>	<i>76-77</i>	<i>78-87</i>
	-0.122	0	0.146	-0.490	0.853	-0.573	0.207	-0.133	0.048	0
	<i>88-91</i>	<i>92-95</i>	<i>96-98</i>	<i>99-101</i>	<i>102-105</i>	<i>106-107</i>	<i>108-117</i>	<i>118-119</i>	<i>120-123</i>	<i>124-127</i>
	0.141	-0.493	0.857	-0.573	0.205	-0.122	0	-0.044	0.144	-0.474
	<i>128</i>									
0.751										
3	<i>1-6</i>	<i>7-14</i>	<i>15-24</i>	<i>25-32</i>	<i>33-40</i>	<i>41-46</i>	<i>47-53</i>	<i>54-64</i>	<i>65-72</i>	<i>73-78</i>
	0.580	-0.279	0.181	-0.304	0.402	-0.502	0.293	-0.185	0.355	-0.472
	<i>79-87</i>	<i>88-96</i>	<i>97-104</i>	<i>105-110</i>	<i>111-117</i>	<i>118-121</i>	<i>122-128</i>			
	0.245	-0.271	0.404	-0.478	0.221	-0.348	0.51			



(a)



(b)



(c)

Figure 4.40: Three-level wavelet zero-crossing representation of the contour of face-3 in Figure 4.21. (a): Level-1 wavelet zero-crossing representation. (b): C: Level-2 wavelet zero-crossing representation. (c): Level-3 wavelet zero-crossing representation.

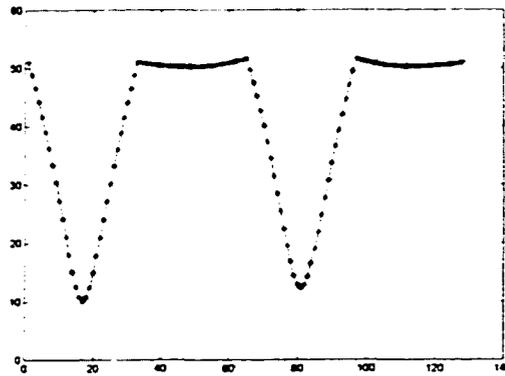


Figure 4.41: Radial distance function for the contour of the spline face-5 in Figure 4.21

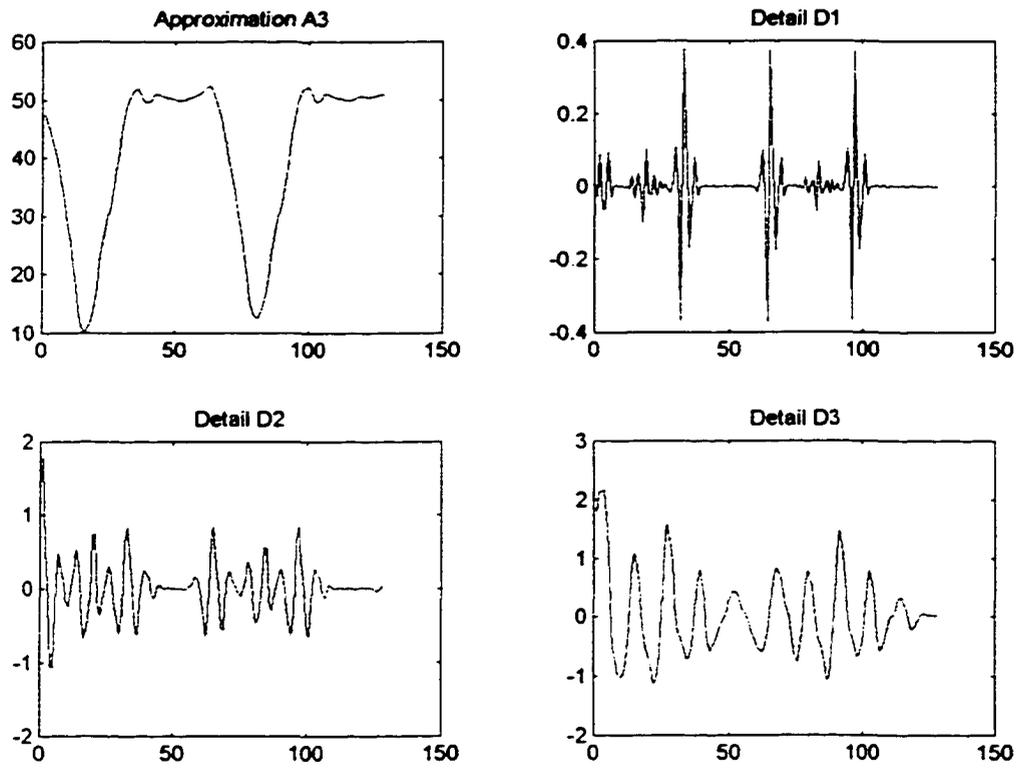
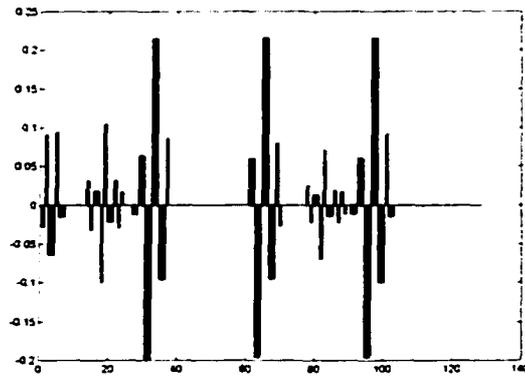


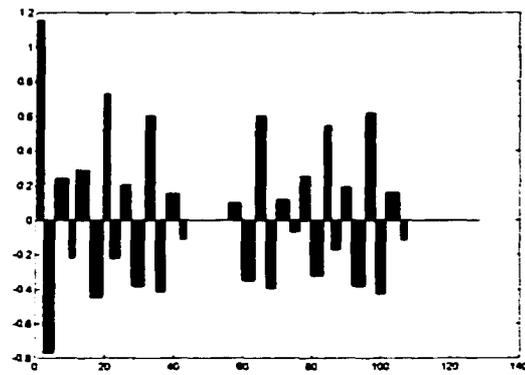
Figure 4.42: Three-level wavelet decomposition of radial distance function in Figure 4.41

Table 4.8: Multi-level coding for the contour of spline face-5 in Figure 4.21

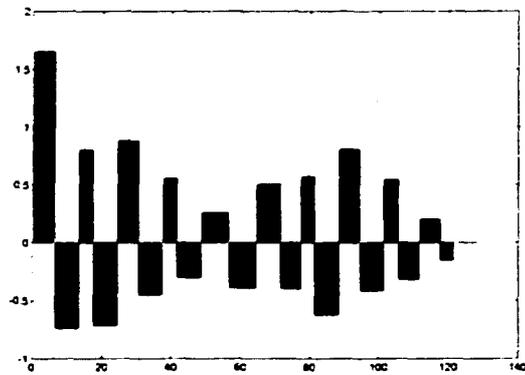
Level	Wavelet Zero-Crossing Representation									
1	<i>1</i>	<i>2</i>	<i>3-4</i>	<i>5</i>	<i>6-7</i>	<i>8-13</i>	<i>14</i>	<i>15</i>	<i>16-17</i>	<i>18</i>
	-0.028	0.091	-0.065	0.094	-0.016	0	0.032	-0.033	0.018	-0.100
	<i>19</i>	<i>20-21</i>	<i>22</i>	<i>23</i>	<i>24</i>	<i>25-26</i>	<i>27-28</i>	<i>29-30</i>	<i>31-32</i>	<i>33-34</i>
	0.105	-0.022	0.032	-0.029	0.017	0	-0.012	0.064	-0.200	0.215
	<i>35-36</i>	<i>37</i>	<i>38-60</i>	<i>61-62</i>	<i>63-64</i>	<i>65-66</i>	<i>67-68</i>	<i>69</i>	<i>70</i>	<i>71-77</i>
	-0.096	0.087	0	0.060	-0.197	0.216	-0.096	0.080	-0.028	0
	<i>78</i>	<i>79</i>	<i>80-81</i>	<i>82</i>	<i>83</i>	<i>84-85</i>	<i>86</i>	<i>87</i>	<i>88</i>	<i>89</i>
	0.027	-0.022	0.013	-0.070	0.071	-0.014	0.019	-0.023	0.017	-0.012
	<i>90</i>	<i>91-92</i>	<i>93-94</i>	<i>95-96</i>	<i>97-98</i>	<i>99-100</i>	<i>101</i>	<i>102-103</i>	<i>104-128</i>	
0	-0.012	0.061	-0.197	0.215	-0.100	0.092	-0.015	0		
2	<i>1-2</i>	<i>3-5</i>	<i>6-9</i>	<i>10-11</i>	<i>12-15</i>	<i>16-19</i>	<i>20-21</i>	<i>22-24</i>	<i>25-27</i>	<i>28-31</i>
	1.157	-0.771	0.244	-0.218	0.289	-0.448	0.730	-0.220	0.205	-0.383
	<i>32-34</i>	<i>35-37</i>	<i>38-41</i>	<i>42-43</i>	<i>44-55</i>	<i>56-59</i>	<i>60-63</i>	<i>64-66</i>	<i>67-69</i>	<i>70-73</i>
	0.605	-0.415	0.156	-0.109	0	0.103	-0.353	0.605	-0.396	0.120
	<i>74-76</i>	<i>77-79</i>	<i>80-83</i>	<i>84-85</i>	<i>86-88</i>	<i>89-91</i>	<i>92-95</i>	<i>96-98</i>	<i>99-101</i>	<i>102-105</i>
	-0.067	0.251	-0.321	0.546	-0.168	0.193	-0.384	0.622	-0.429	0.162
	<i>106-107</i>	<i>108-128</i>								
-0.113	0									
3	<i>1-6</i>	<i>7-13</i>	<i>14-17</i>	<i>18-24</i>	<i>25-30</i>	<i>31-37</i>	<i>38-41</i>	<i>42-48</i>	<i>49-56</i>	<i>57-64</i>
	1.654	-0.739	0.804	-0.713	0.881	-0.449	0.559	-0.301	0.259	-0.391
	<i>65-71</i>	<i>72-77</i>	<i>78-81</i>	<i>82-88</i>	<i>89-94</i>	<i>95-101</i>	<i>102-105</i>	<i>106-111</i>	<i>112-117</i>	<i>118-121</i>
	0.502	-0.397	0.571	-0.624	0.807	-0.412	0.545	-0.315	0.200	-0.151
	<i>122-128</i>									
0										



(a)



(b)



(c)

Figure 4.43: Three-level wavelet zero-crossing representation of the contour of spline face-5 in Figure 4.21. (a): Level-1 wavelet zero-crossing representation. (b): C: Level-2 wavelet zero-crossing representation. (c): Level-3 wavelet zero-crossing representation.

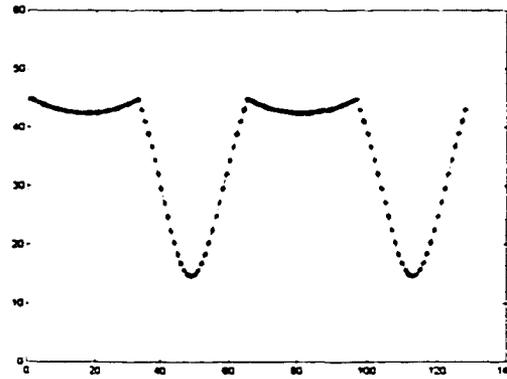


Figure 4.44: Radial distance function for the contour of the cylindrical face-6 in Figure 4.21

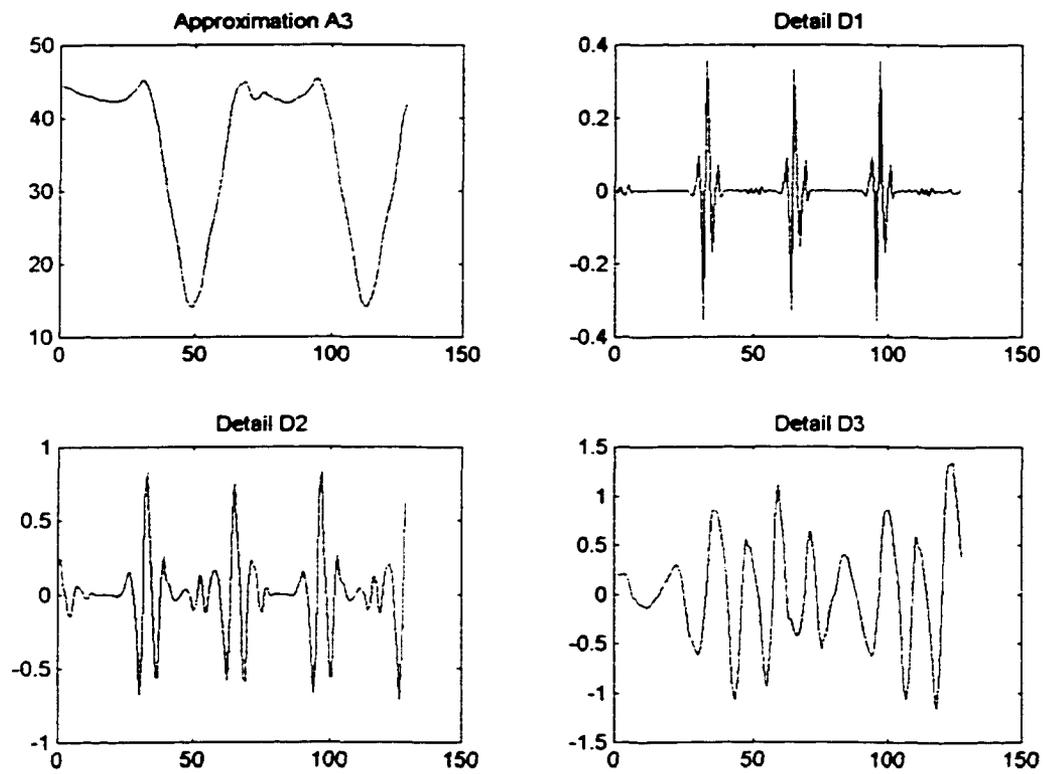
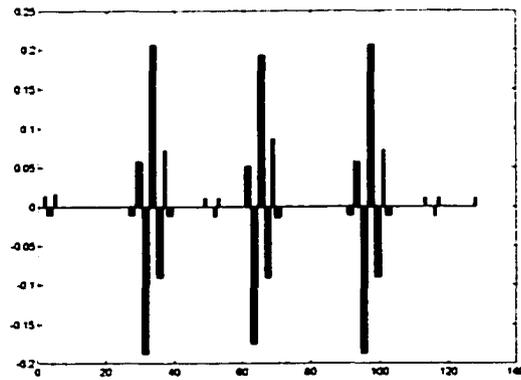


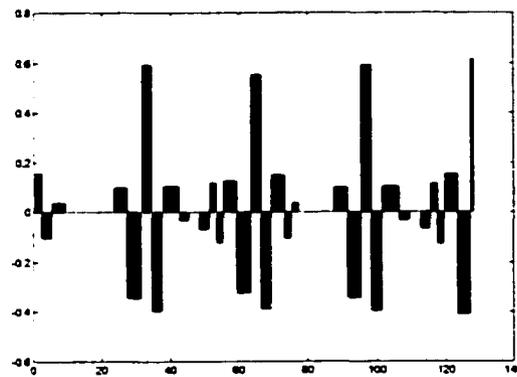
Figure 4.45: Three-level wavelet decomposition of radial distance function in Figure 4.44

Table 4.9: Multi-level coding for the contour of face-6 in Figure 4.21

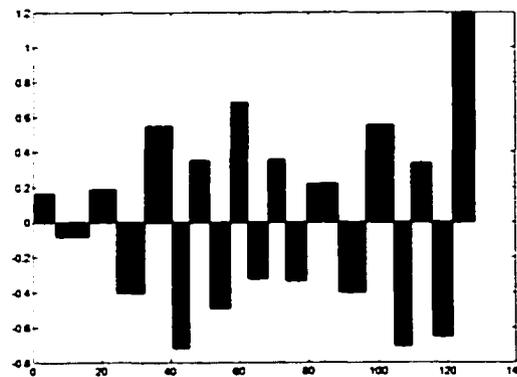
Level	Wavelet Zero-Crossing Representation									
1	<i>1</i>	<i>2</i>	<i>3-4</i>	<i>5</i>	<i>6-26</i>	<i>27-28</i>	<i>29-30</i>	<i>31-32</i>	<i>33-34</i>	<i>35-36</i>
	0	0.014	-0.011	0.017	0	-0.011	0.058	-0.188	0.206	-0.090
	<i>37</i>	<i>38-39</i>	<i>40-48</i>	<i>49</i>	<i>50-51</i>	<i>52</i>	<i>53</i>	<i>54-60</i>	<i>61-62</i>	<i>63-64</i>
	0.073	-0.012	0	0.011	0	-0.013	0.012	0	0.052	-0.175
	<i>65-66</i>	<i>67-68</i>	<i>69</i>	<i>70-71</i>	<i>72-90</i>	<i>91-92</i>	<i>93-94</i>	<i>95-96</i>	<i>97-98</i>	<i>99-100</i>
	0.193	-0.091	0.087	-0.014	0	-0.011	0.058	-0.188	0.206	-0.090
	<i>101</i>	<i>102-103</i>	<i>104-112</i>	<i>113</i>	<i>114-115</i>	<i>116</i>	<i>117</i>	<i>118-127</i>	<i>128</i>	
0.073	-0.012	0	0.011	0	-0.013	0.012	0	0.011		
2	<i>1</i>	<i>3</i>	<i>6</i>	<i>10</i>	<i>24</i>	<i>28</i>	<i>32</i>	<i>35</i>	<i>38</i>	<i>43</i>
	0.156	-0.104	0.037	0	0.098	-0.343	0.591	-0.396	0.104	-0.033
	<i>46</i>	<i>49</i>	<i>52</i>	<i>54</i>	<i>56</i>	<i>60</i>	<i>64</i>	<i>67</i>	<i>70</i>	<i>74</i>
	0	-0.068	0.117	-0.120	0.126	-0.322	0.554	-0.387	0.148	-0.105
	<i>76</i>	<i>78</i>	<i>88</i>	<i>92</i>	<i>96</i>	<i>99</i>	<i>102</i>	<i>107</i>	<i>110</i>	<i>113</i>
	0.037	0	0.098	-0.343	0.591	-0.396	0.104	-0.033	0	-0.067
	<i>116</i>	<i>118</i>	<i>120</i>	<i>124</i>	<i>128</i>					
0.114	-0.126	0.154	-0.409	0.617						
3	<i>1</i>	<i>7</i>	<i>17</i>	<i>25</i>	<i>33</i>	<i>41</i>	<i>46</i>	<i>52</i>	<i>58</i>	<i>63</i>
	0.167	-0.082	0.189	-0.402	0.550	-0.719	0.354	-0.491	0.684	-0.323
	<i>69</i>	<i>74</i>	<i>80</i>	<i>89</i>	<i>97</i>	<i>105</i>	<i>110</i>	<i>116</i>	<i>122</i>	
0.356	-0.336	0.221	-0.401	0.553	0.706	0.334	-0.654	1.196		



(a)



(b)



(c)

Figure 4.46: Three-level wavelet zero-crossing representation of the contour of the cylindrical face-6 in Figure 4.21. (a): Level-1 wavelet zero-crossing representation. (b): C: Level-2 wavelet zero-crossing representation. (c): Level-3 wavelet zero-crossing representation.

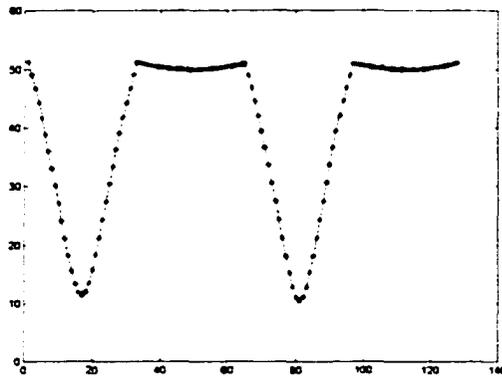


Figure 4.47: Radial distance function for the contour of the spline face-8 in Figure 4.21

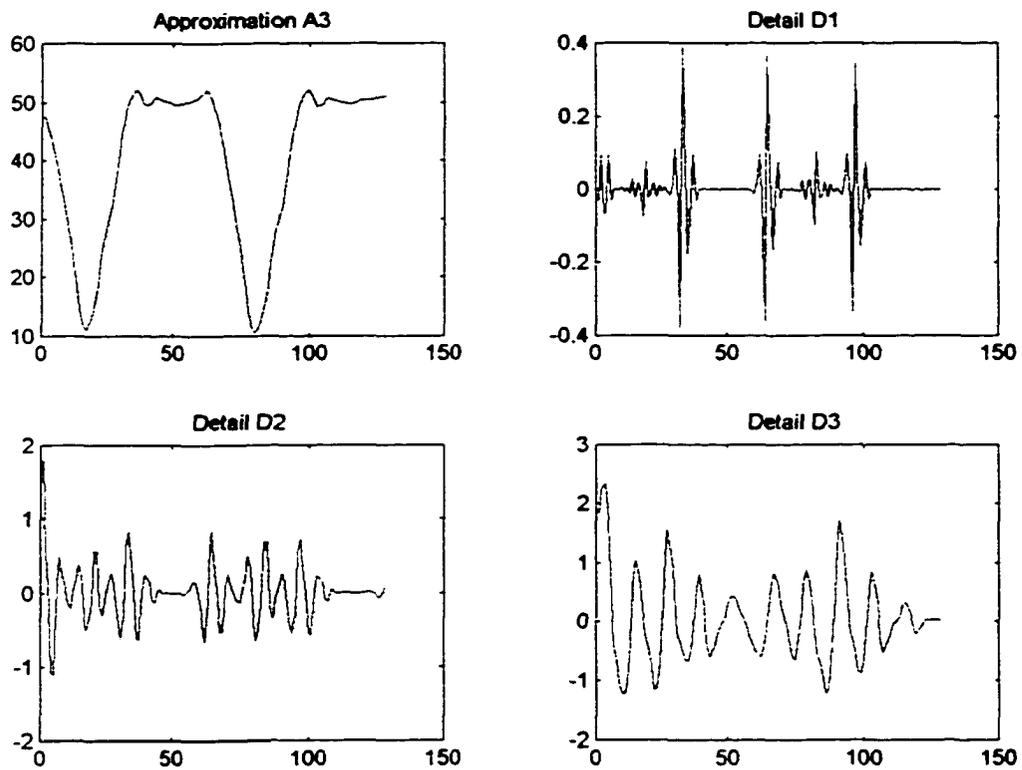
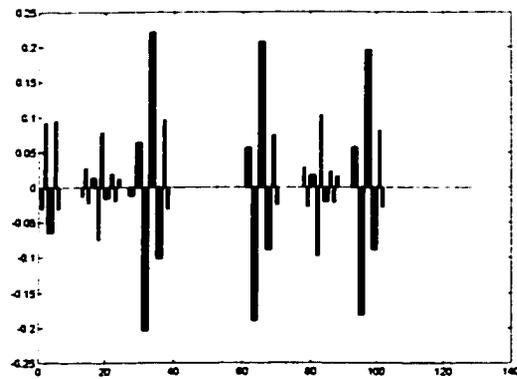


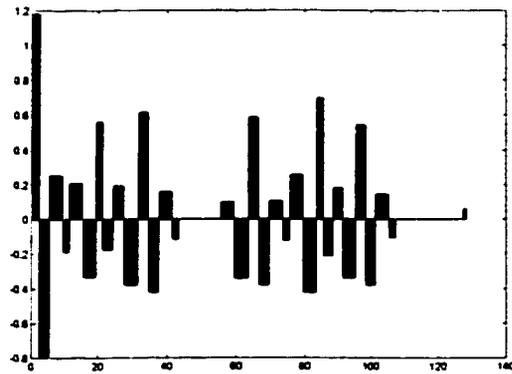
Figure 4.48: Three-level wavelet decomposition of radial distance function in Figure 4.47

Table 4.10: Multi-level coding for the contour of spline face-8 in Figure 4.21

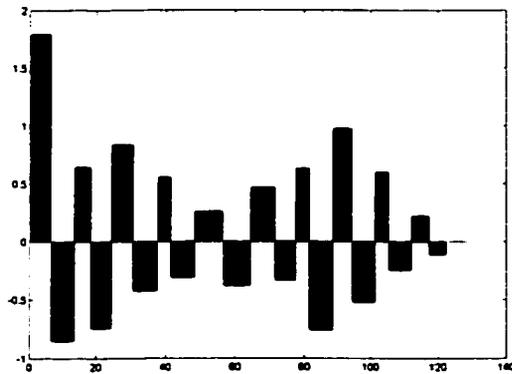
Level	Wavelet Zero-Crossing Representation									
1	<i>1</i>	<i>2</i>	<i>3-4</i>	<i>5</i>	<i>6</i>	<i>7-12</i>	<i>13</i>	<i>14</i>	<i>15</i>	<i>16</i>
	-0.030	0.093	-0.065	0.095	-0.031	0	-0.014	0.027	-0.023	0.014
	<i>18</i>	<i>19</i>	<i>20-21</i>	<i>22</i>	<i>23</i>	<i>24</i>	<i>25-26</i>	<i>27</i>	<i>29-30</i>	<i>31-32</i>
	-0.075	0.078	-0.015	0.019	-0.020	0.012	0	-0.012	0.064	-0.204
	<i>33-34</i>	<i>35-36</i>	<i>37</i>	<i>38</i>	<i>39-60</i>	<i>61-62</i>	<i>63-64</i>	<i>65-66</i>	<i>67-68</i>	<i>69</i>
	0.221	-0.103	0.096	-0.031	0	0.057	-0.190	0.208	-0.090	0.074
	<i>70</i>	<i>71-77</i>	<i>78</i>	<i>79</i>	<i>80-81</i>	<i>82</i>	<i>83</i>	<i>84-85</i>	<i>86</i>	<i>87</i>
	-0.026	0	0.029	-0.028	0.017	-0.098	0.103	-0.020	0.023	-0.022
	<i>88</i>	<i>89-92</i>	<i>93-94</i>	<i>95-96</i>	<i>97-98</i>	<i>99-100</i>	<i>101</i>	<i>102</i>	<i>103-128</i>	
0.015	0	0.058	-0.182	0.196	-0.089	0.081	-0.029	0		
2	<i>1-2</i>	<i>3-5</i>	<i>6-9</i>	<i>10-11</i>	<i>12-15</i>	<i>16-19</i>	<i>20-21</i>	<i>22-24</i>	<i>25-27</i>	<i>28-31</i>
	1.180	-0.791	0.256	-0.187	0.208	-0.332	0.560	-0.174	0.194	-0.379
	<i>32-34</i>	<i>35-37</i>	<i>38-41</i>	<i>42-43</i>	<i>44-55</i>	<i>56-59</i>	<i>60-63</i>	<i>64-66</i>	<i>67-69</i>	<i>70-73</i>
	0.612	-0.422	0.160	-0.115	0	0.098	-0.342	0.585	-0.379	0.108
	<i>74-75</i>	<i>76-79</i>	<i>80-83</i>	<i>84-85</i>	<i>86-88</i>	<i>89-91</i>	<i>92-95</i>	<i>96-98</i>	<i>99-101</i>	<i>102-105</i>
	-0.122	0.261	-0.422	0.698	-0.210	0.184	-0.340	0.542	-0.379	0.146
	<i>106-107</i>	<i>108-127</i>	<i>128</i>							
-0.107	0	0.062								
3	<i>1-6</i>	<i>7-13</i>	<i>14-18</i>	<i>19-24</i>	<i>25-30</i>	<i>31-37</i>	<i>38-41</i>	<i>42-48</i>	<i>49-56</i>	<i>57-64</i>
	1.796	-0.848	0.642	-0.745	0.838	-0.428	0.552	-0.307	0.260	-0.383
	<i>65-71</i>	<i>72-77</i>	<i>78-81</i>	<i>82-88</i>	<i>89-94</i>	<i>95-101</i>	<i>102-105</i>	<i>106-112</i>	<i>113-117</i>	<i>118-122</i>
	0.466	-0.336	0.631	-0.763	0.975	-0.522	0.599	-0.244	0.220	-0.112
	<i>123-128</i>									
0										



(a)



(b)



(c)

Figure 4.49: Three-level wavelet zero-crossing representation of the contour of the spline-8 in Figure 4.21. (a): Level-1 wavelet zero-crossing representation. (b): Level-2 wavelet zero-crossing representation. (c): Level-3 wavelet zero-crossing representation.

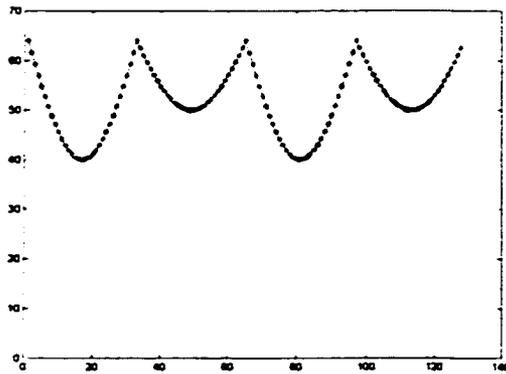


Figure 4.50: Radial distance function for the contour of the face-9 in Figure 4.21

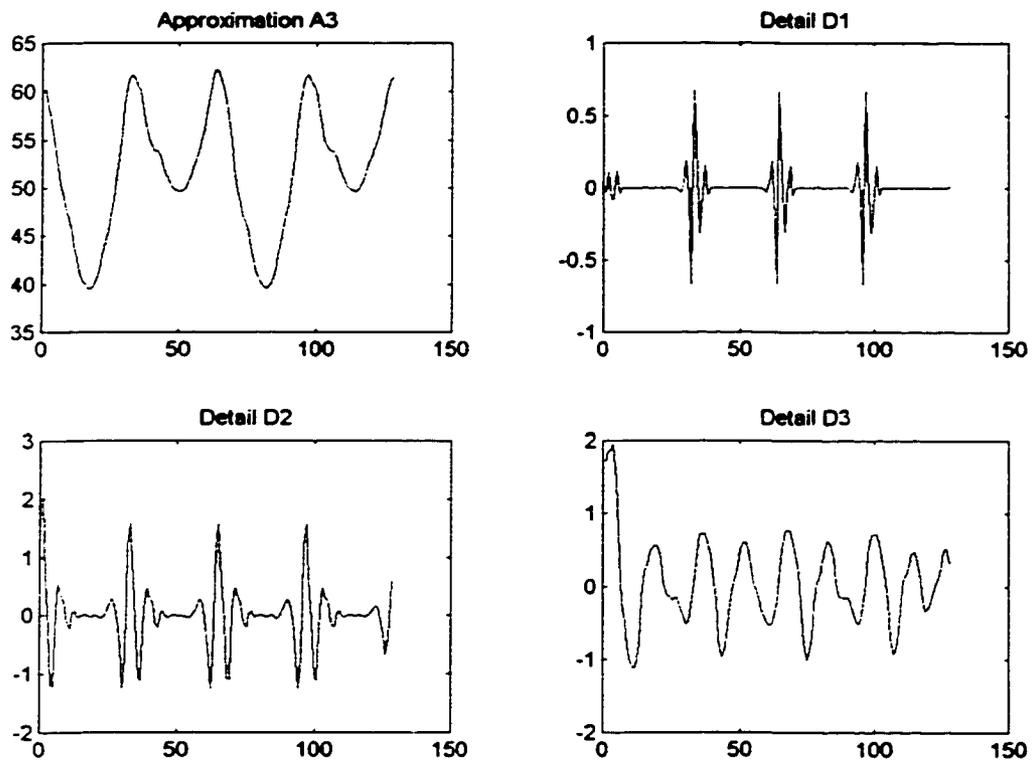
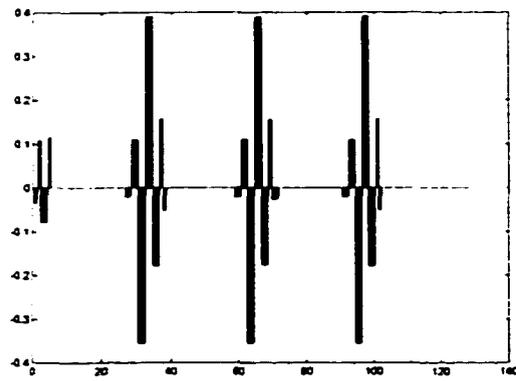


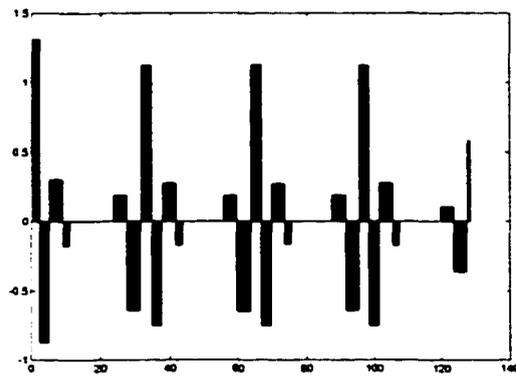
Figure 4.51: Three-level wavelet decomposition of radial distance function in Figure 4.50

Table 4.11: Multi-level coding for the contour of the face-9 in Figure 4.21

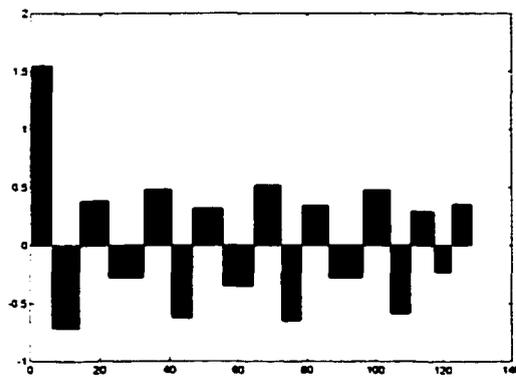
Level	Wavelet Zero-Crossing Representation									
1	<i>1</i>	<i>2</i>	<i>3-4</i>	<i>5</i>	<i>6-26</i>	<i>27-28</i>	<i>29-30</i>	<i>31-32</i>	<i>33-34</i>	<i>35-36</i>
	-0.035	0.108	-0.078	0.115	0	-0.021	0.110	-0.355	0.390	-0.178
	<i>37</i>	<i>38</i>	<i>39-58</i>	<i>59-60</i>	<i>61-62</i>	<i>63-64</i>	<i>65-66</i>	<i>67-68</i>	<i>69</i>	<i>70-71</i>
	0.157	-0.051	0	-0.022	0.111	-0.355	0.389	-0.176	0.155	-0.025
	<i>72-90</i>	<i>91-92</i>	<i>93-94</i>	<i>95-96</i>	<i>97-98</i>	<i>99-100</i>	<i>101</i>	<i>102</i>	<i>103-128</i>	
0	-0.021	0.110	-0.355	0.390	-0.178	0.157	-0.051	0		
2	<i>1</i>	<i>3</i>	<i>6</i>	<i>10</i>	<i>12</i>	<i>24</i>	<i>28</i>	<i>32</i>	<i>35</i>	<i>38</i>
	1.312	-0.875	0.302	-0.177	0	0.189	-0.645	1.124	-0.754	0.273
	<i>42</i>	<i>44</i>	<i>56</i>	<i>60</i>	<i>64</i>	<i>67</i>	<i>70</i>	<i>74</i>	<i>76</i>	<i>88</i>
	-0.174	0	0.185	-0.647	1.127	-0.754	0.271	-0.165	0	0.189
	<i>92</i>	<i>96</i>	<i>99</i>	<i>102</i>	<i>106</i>	<i>108</i>	<i>120</i>	<i>124</i>	<i>128</i>	
-0.645	1.124	-0.754	0.273	-0.174	0	0.106	-0.363	0.581		
3	<i>1</i>	<i>7</i>	<i>15</i>	<i>23</i>	<i>33</i>	<i>41</i>	<i>47</i>	<i>56</i>	<i>65</i>	<i>73</i>
	1.547	-0.712	0.382	-0.278	0.479	-0.622	0.317	-0.347	0.516	-0.648
	<i>79</i>	<i>87</i>	<i>97</i>	<i>105</i>	<i>111</i>	<i>118</i>	<i>123</i>			
0.341	-0.280	0.477	-0.584	0.293	-0.231	0.355				



(a)



(b)



(c)

Figure 4.52: Three-level wavelet zero-crossing representation of the contour of the face-9 in Figure 4.21. (a): Level-1 wavelet zero-crossing representation. (b): C: Level-2 wavelet zero-crossing representation. (c): Level-3 wavelet zero-crossing representation.

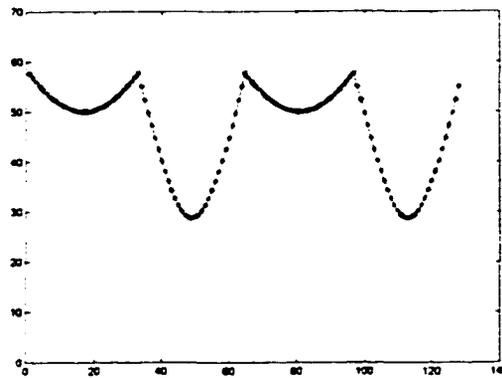


Figure 4.53: Radial distance function for the contour of the face-10 in Figure 4.21

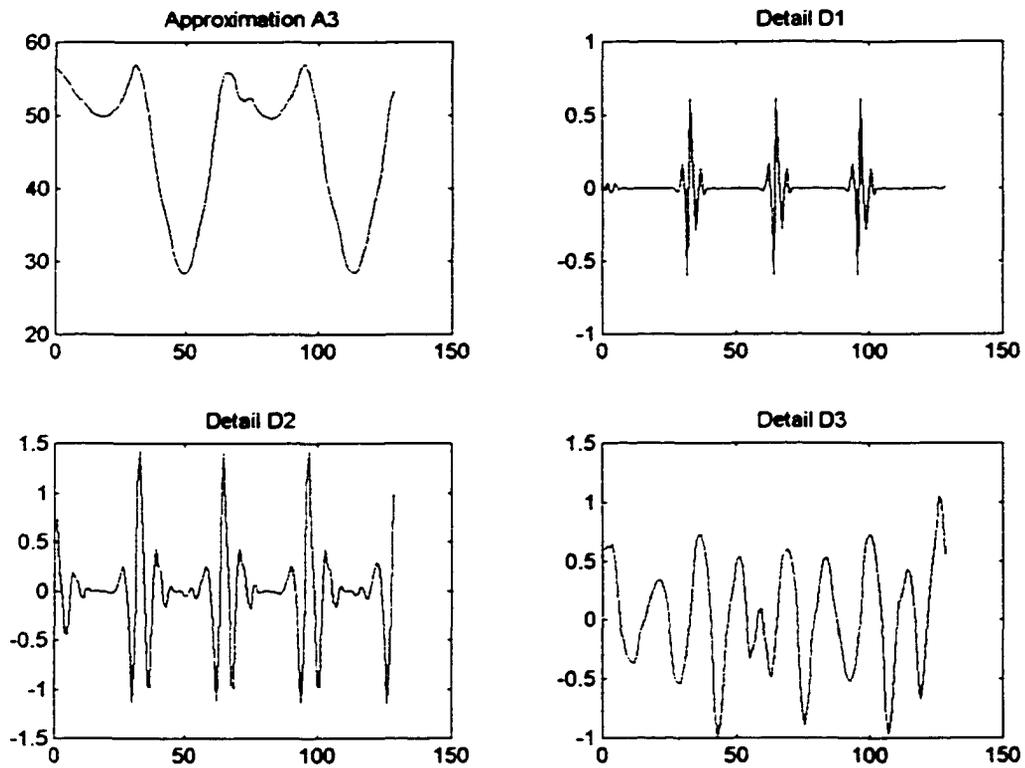
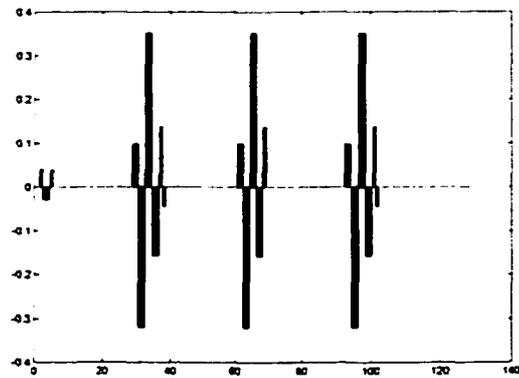


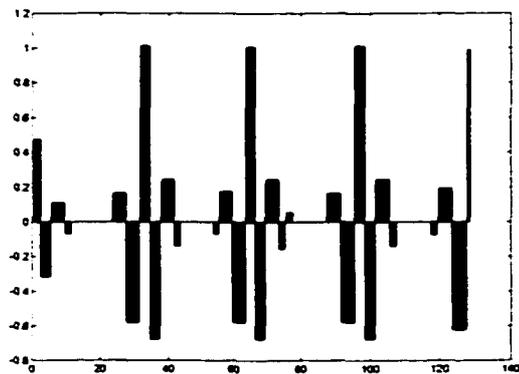
Figure 4.54: Three-level wavelet decomposition of radial distance function in Figure 4.53

Table 4.12: Multi-level coding for the contour of the face-10 in Figure 4.21

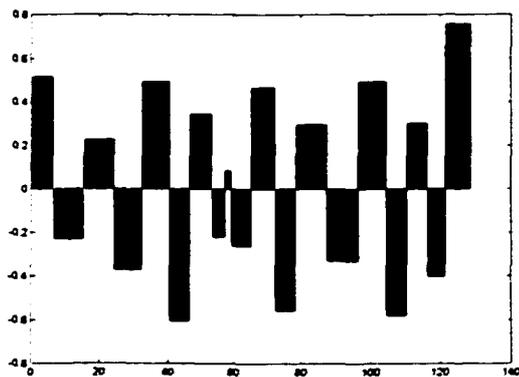
Level	Wavelet Zero-Crossing Representation									
1	<i>1</i>	<i>2</i>	<i>3-4</i>	<i>5</i>	<i>6-28</i>	<i>29-30</i>	<i>31-32</i>	<i>33-34</i>	<i>35-36</i>	<i>37</i>
	0	0.041	-0.028	0.040	0	0.098	-0.320	0.350	-0.158	0.137
	<i>38</i>	<i>39-60</i>	<i>61-62</i>	<i>63-64</i>	<i>65-66</i>	<i>67-68</i>	<i>69</i>	<i>70-92</i>	<i>93-94</i>	<i>95-96</i>
	-0.045	0	0.099	-0.321	0.351	-0.158	0.137	0	0.098	-0.320
	<i>97-98</i>	<i>99-100</i>	<i>101</i>	<i>102</i>	<i>103-128</i>					
0.350	-0.158	0.137	-0.045	0						
2	<i>1</i>	<i>3</i>	<i>6</i>	<i>10</i>	<i>12</i>	<i>24</i>	<i>28</i>	<i>32</i>	<i>35</i>	<i>38</i>
	0.456	-0.316	0.110	-0.069	0	0.165	-0.582	1.013	-0.678	0.242
	<i>42</i>	<i>44</i>	<i>54</i>	<i>56</i>	<i>60</i>	<i>64</i>	<i>67</i>	<i>70</i>	<i>74</i>	<i>76</i>
	-0.138	0	-0.067	0.178	-0.579	1.006	-0.677	0.245	-0.155	0.054
	<i>78</i>	<i>88</i>	<i>92</i>	<i>96</i>	<i>99</i>	<i>102</i>	<i>106</i>	<i>108</i>	<i>118</i>	<i>120</i>
	0	0.165	-0.582	1.013	-0.678	0.242	-0.138	0	-0.073	0.195
	<i>124</i>	<i>128</i>								
	-0.626	0.990								
3	<i>1</i>	<i>7</i>	<i>16</i>	<i>25</i>	<i>33</i>	<i>41</i>	<i>47</i>	<i>54</i>	<i>58</i>	<i>60</i>
	0.516	-0.230	0.225	-0.372	0.491	-0.606	0.344	-0.221	0.084	-0.263
	<i>66</i>	<i>73</i>	<i>79</i>	<i>88</i>	<i>97</i>	<i>105</i>	<i>111</i>	<i>117</i>	<i>122</i>	
0.467	-0.558	0.295	-0.331	0.494	-0.579	0.301	-0.401	0.755		



(a)



(b)



(c)

Figure 4.55: Three-level wavelet zero-crossing representation of the contour of the face-10 in Figure 4.21. (a): Level-1 wavelet zero-crossing representation. (b): C: Level-2 wavelet zero-crossing representation. (c): Level-3 wavelet zero-crossing representation.

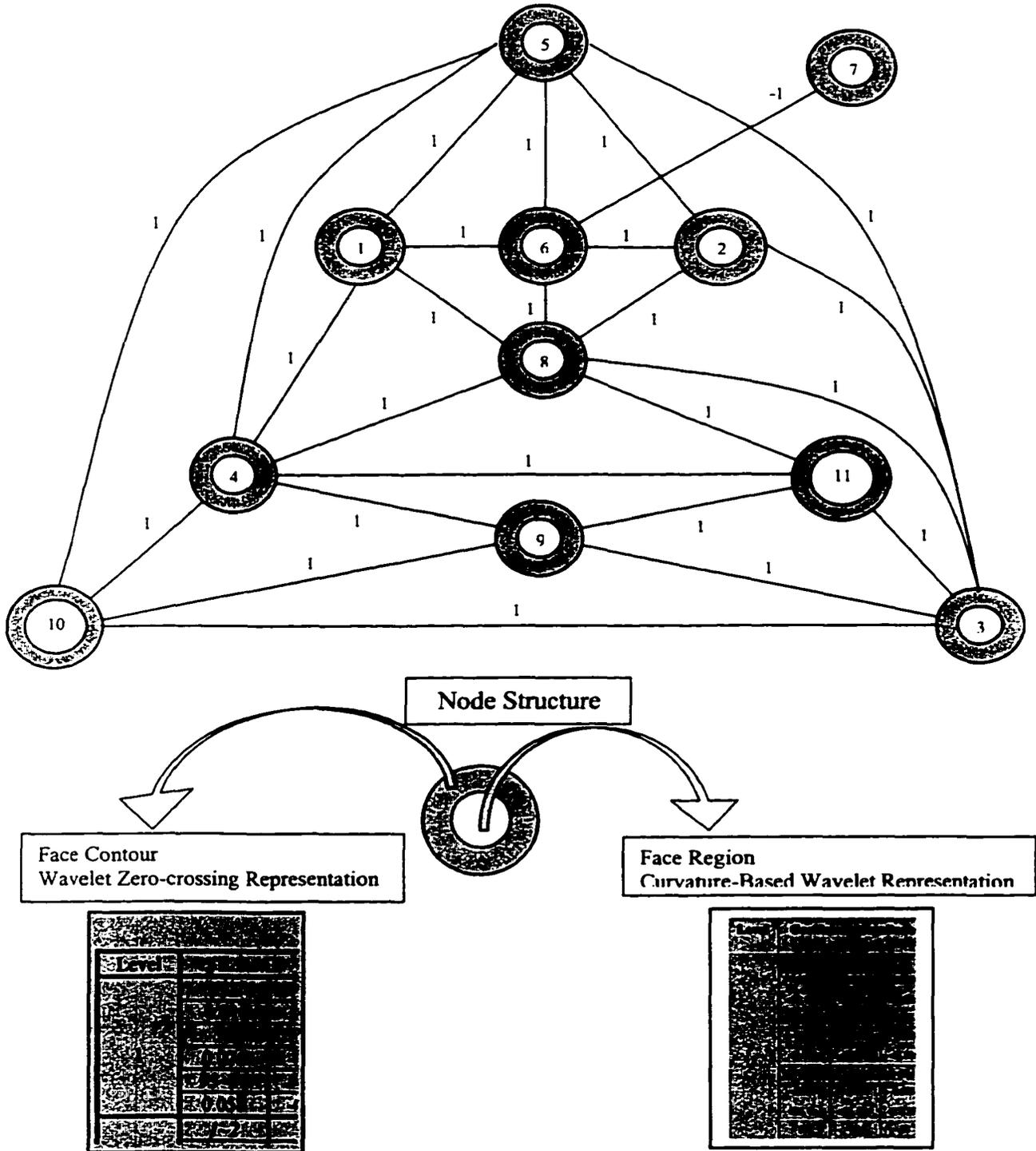


Figure 4.56: The graphical representation of model in Figure 4.21 with face region and contour coding. Dotted circles and heavy solid lines correspond to face clusters by surface ranking algorithm in Chapter 3.

4.6 Summary

This chapter has addressed the development of a transform-invariant coding system for CAD models by multi-scale wavelet representations. The coding system is implemented in terms of the topology entity—face. A face consists of the internal content (surface patch) and the external closed contour (contour). Both the internal region and external boundary are chosen for coding purpose.

Discrete wavelet transforms (DWT) are unstable under the translations, rotations and dilations of the input shape signal. In order to overcome this limitation, one of the most intrinsic shape characteristics—curvature, which is invariant to transform and parameterization, is used to represent the face region. The face boundary is approximated by the radial distance function.

Wavelet transforms are used to represent these two transform-invariant shape descriptors in terms of fine-to-coarse approximations and details. A few of these wavelet approximation and detail coefficients are then selected and normalized to be codes of the corresponding face.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

6.1 Summary and Conclusions

The general objective of this research work is to develop design-for-manufacturing (DFM) tools to automate the verification of Design for Manufacturing (DFM) rules. This dissertation has addressed the development of a generalized framework for high-level geometric representations of CAD models and form features to automate algorithmic search and retrieval of manufacturing information. A new wavelet-based ranking algorithm is developed to generate surface-based representations as input for the extraction of form features with non-planar surfaces in CAD models. A transform-invariant coding system for CAD models by multi-scale wavelet representations is also presented. The coding procedure is based on both the internal regions and external contours of topology entities – faces.

The objective of using a wavelet-based shape analysis approach is to overcome the main limitation of the alternative feature extraction approaches, namely their restriction to planar surfaces or simple curved surfaces. In this work, a CAD model is discretized by a regular grid and represented in a binary three-dimensional matrix. In this binary representation, form features can then be interpreted as singularities in a product's "shape signal." A singularity is an abrupt change in a shape signal.

Discrete wavelet transform (DWT) is used in this research for the shape analysis of the digital representation. DWT can decompose a signal into a set of orthogonal, spatially oriented representations within certain frequency intervals. It approximates a cell value by a

weighed average of its neighboring cells. The difference between the actual cell value and the predicted value is called the detail of the cell. When WT is applied to the product's shape signal, it can detect the sharp contrast caused by a feature's surfaces and assign different detail coefficients according to the surfaces' curvature patterns.

However DWT is unstable under the translations, rotations and dilations of the input shape signal. In order to overcome this limitation and find a transform-invariant representation for CAD models, the face region is represented by its curvatures and the face contour is approximated by the radial distance function. DWT then decompose these two transform-invariant shape descriptors into approximations and details.

This dissertation contributes to the Design for Manufacturing (DFM) research discipline by using wavelet mathematical tools for CAD shape analysis and representation and providing feasible implementation procedures. The detailed contributions of this dissertation are as follows:

1. The definition of form features as singularities in an object's shape signal. A form feature is a region of abrupt change in the shape of a product and has a characteristic pattern of variation in terms of its surfaces' curvatures. It is a novel concept and the core for the proposed ranking and clustering algorithms.

2. The definition of feature primitives – faces. A face is the boundary of transition between the inside and outside of the product. As a result, a surface is presented as a sharp contrast in the binary representation. A form feature can therefore be characterized by the singularity pattern of its surfaces. For the purpose of computational efficiency, a face that has the highest ranking in the ranking and clustering algorithms is selected as “seed surface”. The seed surface is used as a starting point for surface grouping into form features.

3. The application of wavelet transforms for multi-resolution shape analysis. Digital image shape analysis is defined as the process of characterizing shapes and developing shape similarity measures for images. Within the context of CAD models, shape analysis is defined as the process of coding and classification of surfaces. Wavelet transforms can decompose a three-dimensional signal into a set of orthogonal, spatially oriented representations. It approximates a cell value by a weighed average of its neighboring cells. The difference between the actual cell value and the predicted value is called the detail of the cell. When wavelet transform is applied to the three-dimensional binary matrix of an object, it can detect the sharp contrast caused by a feature's surfaces and assign different detail coefficients according to the surfaces' curvature patterns.

4. The development of surface ranking algorithm. Surfaces in the CAD model are analyzed and classified based upon their three-dimensional wavelet detail coefficients. Once these wavelet coefficients are segmented into intervals relative to their maximum, surfaces can then be ranked in terms of its coefficients' variation pattern. Finally form features are extracted by clustering the ranked surfaces.

5. The development of face region coding algorithm. This segment of the research attempts to find transform-invariant descriptions for face regions, which are essential for developing a coding and classification system for geometric models. One of the most intrinsic shape characteristics – curvature, which is invariant to transform and parameterization, is selected to represent the face region. Wavelet transforms are used to represent this curvature-based shape descriptor in terms of fine-to-coarse approximations and details. A few of these wavelet approximation and detail coefficients are then selected and normalized to be codes of the corresponding face region.

6. The development of face loop coding algorithm by wavelet zero-crossing representation. The main idea of this part of the research is to represent face loops by fine-to-coarse approximations and details based on the wavelet transform zero-crossing representation. In order to build the representation, a radial function describing the face loop is constructed, and its wavelet transform zero-crossing representation is generated.

6.2 Future Research Recommendations

This dissertation has focused on developing the prototype of high-level geometric representations of CAD models and form features by wavelet transforms. There are two aspects in the future research development of the automated DFM tools.

The first is the extension of current research presented in this dissertation by developing a generic structure for modeling the capabilities of manufacturing processes. The wavelet-based face representations, proposed as input for form feature extraction procedures, have been specified to give room for the inclusion of additional manufacturing resource information. Once manufacturing resource information is available in the wavelet-based face representations, the research can move to the feature matching and recognition stage.

The second is the two fundamental research issues related to wavelet bases. They are generating transform-invariant wavelet bases and setting up irregular sampling theorem for wavelet bases. As mentioned in Chapter 4, one of the major drawbacks of wavelet transforms is their lack of invariance under translation, dilation and rotation of the input signal. This problem has been addressed by some ongoing researches and solutions have been proposed [Freeman and Adelson 1991, Simoncelli et al. 1992]. Another drawback of

wavelet bases is that they can not handle irregularly sampled signals correctly. Chen et al. (1997) addressed this problem and a new irregular wavelet scheme in initial stage was proposed.

BIBLIOGRAPHY

- [1] K. Arbter, W.E. Snyder, H. Burkhardt, and G. Hirzinger. "Application of Affine-Invariant Fourier Descriptors to Recognition of 3D Objects." *IEEE Trans. Pattern Analysis and Machine Intelligence* 12.7 (1990): 640-646.
- [2] J.M. Beck, R.T. Farouki, and J.K. Hinds. "Surface Analysis Methods." *IEEE Computer Graphics and Applications* Dec. 1986: 18-36.
- [3] A. Belyaev and Y. Ohtake. "An Image Processing Approach to Detection of Ridges and Ravines on Polyhedral Surfaces." Technical Reports, University of Aizu, Japan.
- [4] P. Besl and R.C. Jain. "Invariant Surface Characteristics for 3D Object Recognition in Range Images." *Computer Vision, Graphics and Image Processing* 33 (1986): 33-80.
- [5] P.J. Burt and E.H. Adelson. "The Laplacian Pyramid as a Compact Image Code." *IEEE Trans. on Communications* 31.4 (1983): 532-540.
- [6] C.H. Chen, J.S. Lee, and Y.N. Sun. "Wavelet Transformation for Gray-Level Corner Detection." *Pattern Recognition* 28.6 (1995): 853-861.
- [7] W. Chen, S. Itoh, and J. Shiki. "Irregular Sampling Theorems for Wavelet Subspaces." Technical Reports, Department of Information Network Sciences, University of Electro-Communications, Chofu, Tokyo, Japan.
- [8] B.K. Choi, M.M. Barash, and D.C. Anderson. "Automatic Recognition of Machined Surface from a 3D Solid Model." *Computer-Aided Design* 16.2 (1984): 81-86.
- [9] S.H. Chuang and M.R. Henderson. "Three-dimensional Shape Pattern Recognition Using Vertex Classification and Vertex-edge Graphs." *Computer-Aided Design* 22.6 (1990): 377-387.
- [10] C. K. Chui. *An Introduction to Wavelets, Wavelet Analysis and Its Applications*. Vol. 1. San Diego: Academic Press, 1992.
- [11] C.K. Chui. *Wavelets: A Mathematical Tool for Signal Processing*. Society for Industrial and Applied Mathematics, 1997.
- [12] R.J. Clarke. *Transform Coding of Images*. London: Academic Press, 1985.
- [13] J Corney and D E Clark. "Method for Finding Holes and Pockets that Connect Multiple Faces in 2 1/2D Objects." *Computer-Aided Design* 23.10 (1991): 658-669.

- [14] I. Daubechies. "Orthonormal Bases of Compactly Supported Wavelets." *Communications on Pure and Applied Mathematics* 41 (1988): 909-966.
- [15] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF, Regional Conference Series in Applied Mathematics, SIAM 1, 1992.
- [16] I. Daubechies. "Orthonormal Bases of Compactly Support Wavelet II: Variations on Theme" *SIAM Journal on Mathematical Analysis* 24.2 (1993): 499-519.
- [17] R.A. Devore, B. Lucier, and Z. Yang. "Feature Extraction in Digital Mammography." *Wavelets in Medicine and Biology*. Ed. A. Aldroubi and M. Unser. Boca Raton, Florida: CRC Press, 1996. 145-161.
- [18] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach. "Ridges for Image Analysis." *Journal of Mathematical Imaging and Vision* 4 (1994): 353-373.
- [19] D. Eberly. *Ridges in Image and Data Analysis*. Kluwer, 1996.
- [20] M.P. Do Carmo. *Differential Geometry of Curves and Surfaces*. New Jersey: Prentice-Hall, Inc., 1976.
- [21] G. Eichmann and M. Jankowski. "Fourier Shape Descriptors for Surfaces of Multi-Dimensional Closed Volumes." *Proceedings of SPIE: Intelligent Robots and Computer Vision* 579 (1985): 482-487.
- [22] T.J. Fan, G. Medioni, and R. Nevatia. "Segmented Descriptions of 3-D Surfaces." *IEEE Journal of Robotics and Automation* RA-3.6 (1987): 527-538.
- [23] T.J. Fan, G. Medioni, and R. Nevatia. "Recognizing 3-D Objects Using Surface Descriptions." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (1989): 1140-1157.
- [24] L. De Floriani. "A Graph-based Approach to Object Feature Recognition." *Proceedings of the Third ACM Symposium on Computational Geometry* June 1987: 100-109.
- [25] W.T. Freeman and E.H. Adelson. "The Design and Use of Steerable Filters." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.9 (1991): 891-906.
- [26] R. Gadh and F. Prinz. "Recognition of Geometric Forms Using the Differential Depth Filter." *Computer-Aided Design* 24.11 (1992): 583-598.
- [27] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1997.

- [28] M. H. GroB, R. Koch, L. Lippert, and A. Dreyer. "Multi-Scale Image Texture Analysis in Wavelet Spaces." *IEEE International Conferences on Image Processing* 3 (1994): 412-417.
- [29] J H Han, W C. Regli, and D. Rosen. "Special Panel Session for Feature Recognition" *1997 ASME Design Engineering Technical Conferences*. Sept. (1997): 14-17.
- [30] M.R. Henderson and D.C. Anderson. "Computer Recognition and Extraction of Form Features: A CAD/CAM Link." *Computers in Industry* 5 (1984): 329-339.
- [31] M.R. Henderson, G. Srinath, R. Stage, K. Walker and W. Regli. *Boundary Representation-based Feature Identification, Advances in Feature Based Manufacturing*. Eds. J.J. Shah, M. Mantyla, and D.S. Nau. Amsterdam: Elsevier Science, 1994. 15-38.
- [32] M. Higashi, T. Saitoh, Y. Watanabe, and Y. Watanabe. "Analysis of Aesthetic Free-Form Surfaces by Surface Edges." *Conference on Computer Graphics and Applications*. Seoul, Korea: Pacific Graphics, 1995. 294-305.
- [33] M. Higashi, F. Torihara, N. Takeuchi, T. Sata, T. Saitoh, and M. Hosaka. "Robust Algorithms for Face-Based Representations." *Computer-Aided Design* 29.2 (1997): 135-146.
- [34] C.M. Hoffman. *Geometric and Solid Modeling: An Introduction*. San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1989.
- [35] M. Hosaka. *Modeling of Curves and Surfaces in CAD/CAM*. Berlin: Springer-Verlag, 1992.
- [36] R. Hummel and R. Moniot. "Reconstruction from Zero-crossings in Scale Space." *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.12 (1989): 2111-2130.
- [37] S. Joshi and T.C. Chang. "Graph-Based Heuristics for Recognition of Machined Features from a 3D Solid Model." *Computer-Aided Design* 20.2 (1988): 58-66.
- [38] Gerald Kaiser. *A Friendly Guide to Wavelets*. Boston: BirkhTMuser, 1994.
- [39] Y.S. Kim and D.J. Wilde. "A Convex Decomposition Using Convex Hulls and Local Cause of its Non-Convergence." *ASME Transactions, Journal of Mechanical Design* 114 (1992): 459-476.
- [40] R. Klette, A. Rosenfeld, and F. Sloboda. *Advances in Digital and Computational Geometry*. Singapore: Springer, 1998.

- [41] L Kyprianou. *Shape Classification in Computer-Aided Design*, Ph.D. Thesis. University of Cambridge, 1980.
- [42] T. Laakko and M. Mantyla. "Feature Modeling by Incremental Feature Recognition." *Computer-Aided Design* 25.8 (1993): 479-492.
- [43] D.E. LaCourse. *Handbook of Solid Modeling*. New York: McGraw-hill, Inc., 1995.
- [44] V. Lang, A.G. Belyaev, I.A. Bogaevski, and T.L. Kunii. "Fast Algorithms for Ridge Detection." *Proceedings of 1997 International Conference on Shape Modeling and Applications*. 3-6 Mar. 1997: 189-197.
- [45] Y.C. Lee and K.S. Fu. "Machine Understanding of CSG: Extraction and Unification of Manufacturing Features." *IEEE Computer Graphics and Applications* 7.1 (1987): 20-32.
- [46] G. Little, R. Tuttle, D.E.R. Clark, and J. Corney. "A Feature Complexity Index." *Proceedings Institution of Mechanical Engineers* 212 (1998): 405-412.
- [47] L.G. Miller. *Concurrent Engineering Design*. Society of manufacturing engineers, 1993.
- [48] S. G. Mallat. "A Theory for Multi-Resolution Signal Decomposition: the Wavelet Representation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1.7 (1989): 674-693.
- [49] S. G. Mallat. "Zero-Crossings of a Wavelet Transform." *IEEE Transactions on Information Theory* 37.4 (1991): 1019-1033.
- [50] S. G. Mallat and S. Zhong. "Characterization of Signals from Multi-scale Edges." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.7 (1992): 710-732.
- [51] S. G. Mallat. *A Wavelet Tour of Signal Processing*. San Diego: Academic Press, 1998.
- [52] M. Mantyla. *An Introduction to Solid Modeling*. Rockville, Maryland: Computer Science Press, 1988.
- [53] D. Marr. *Vision*. San Francisco, CA: W.H. Freeman and Co., 1982.
- [54] G. Medioni and R. Nevatia. "Description of 3-D Surfaces Using Curvature Properties." *Proceedings of Image Understanding Workshop, DARPA* Oct. 1984: 291-299.

- [55] S. Muraki. "Volume Data and Wavelet Transforms." *IEEE Computer Graphics & Applications* July 1993: 50-56.
- [56] T. Pavlidis. *Algoirhms for Graphics and Image Processing*. Berlin: Springer-Verlag, 1982.
- [57] E. Persoon and K.S. Fu. "Shape Discrimination Using Fourier Descriptors." *IEEE Transactions on Systems, Man and Cybernetics* SMC-7.3 (1977): 170-179.
- [58] I.R. Porteous. *Geometirc Differentiation for the Intelligence of Curves and Surfaces*. New York: Cambridge University Press, 1994.
- [59] L. Prasad and S.S. Iyengar. *Wavelet Analysis with Applications to Image Processing*. Boca Raton: CRC Press, 1997.
- [60] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. flannery. *Numerical Recipes in C: The Art of Scientific Computing. 2nd Ed.* New York: Cambridge University Press, 1992.
- [61] A.Z. Qamhiyah. "Form Feature Extraction and Coding for Design by Features." Ph.D. Dissertation, Department of Mechanical and Industrial Engineering, University of Toronto, 1996.
- [62] A.Z. Qamhiyah, R.D. Venter and B. Benhabib. "Geometric Reasoning for the Extraction of Form Features." *Computer-Aided Design* 28.11 (1996): 887-903.
- [63] T. Rando and J.A. Roulier. "Designing Faired Parametric Surfaces." *Computer-Aided Design* 23.7 (1991): 492-497.
- [64] W.C. Regli and M.J. Pratt. "What Are Feature Interactions?" *1996 ASME Design Engineering Technical Conferences* Aug. (1996): 18-22.
- [65] O. Rioul and M. Vetterli. "Wavelet and Signal Processing." *IEEE Signal Processing Magzaine* 8.4 (1991): 14 -38.
- [66] H. Sakurai and D.C. Gossard. "Recognizing Shape Features in Solid Models." *IEEE Computer Graphics & Applications* Sept. 1990: 22-32.
- [67] H. Samet. "The Quadtree and Related Hierarchical Data Structures." *ACM Computing Surveys* 16.3 (1984): 287-260.
- [68] J.J. Shah and M.T. Rogers. "Functional Requirements and Conceptual Design of the Feature-Based Modeling System." *Computer-Aided Engineering Journal* 5.1 (1988): 233-241.

- [69] J.J. Shah and M. MTMntyTM. *Parametric and Feature Based CAD/CAM*. New York: John Wiley & Sons, 1995.
- [70] E.P. Simoncelli, W.T. Freeman, E.H. Adelson, and D.J. Heeger. "Shiftable Multiscale Transforms." *IEEE Transactions on Information Theory* 38.2 (1992): 587-607.
- [71] R. Sonthi and R. Gadh. "MMCS and PPCS as Constructs of Curvature Regions for Form Feature Determination." *1997 ASME Design Engineering Technical Conferences* Sept. 1997: 14-17.
- [72] S.M. Staley, M.R. Henderson, and D.C. Anderson. "Using Syntactic Pattern Recognition to Extract Feature Information from a Solid Geometric Data Base." *ASME Computers in Mechanical Engineering* 2.2 (1983): 61-66.
- [73] R. Stage, C. Roberts, and M. Henderson. "Generating Resource Based Flexible Form Manufacturing Features through Objective Driven Clustering." *Computer-Aided Design* 31 (1999): 119-130.
- [74] S. Subrahmanyam and M. Wozny. "An Overview of Automatic Feature Recognition Techniques for Computer-Aided Process Planning." *Computer in Industry* 26 (1995): 1-21.
- [75] M.D. Swanson and A.H. Tewfik. "A Binary Wavelet Decomposition of Binary Images." *IEEE Transactions on Image Processing* 5.12 (1996): 1637-1650.
- [76] B. Telfer and H.H. Szu. "New Wavelet Transform Normalization to Remove Frequency Bias." *Optical Engineering* 31.9 (1992): 1830-1834.
- [77] A. Teolis. *Computational Signal Processing with Wavelets*. Boston: Birkhauser, 1998.
- [78] A.H. Tewfik and P.E. Jorgensen. "On the Choice of a Wavelet for Signal Coding and Processing." *Proceedings of ICASSP'91* 4 (1991): 2015-2028.
- [79] J.P. Thirion and A. Gourdon. "The 3D Marching Lines Algorithm." *Graphical Models and Image Processing* 58.6 (1996): 503-509.
- [80] Q.M. Tieng and W.W. Boles. "Recognition of 2D Object Contours Using the Wavelet Transform Zero-Crossing Representation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.8 (1997): 910-916.
- [81] Q. M. Tieng and W.W. Boles. "Wavelet Based Affine Invariant Representation: a Tool for Recognizing Planar Objects in 3D Space." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.8 (1997): 846-857.

- [82] P.J. van Otterloo. *A Contour -Oriented Approach to Shape Analysis*. London: Prentice Hall, 1991.
- [83] E. Wang and Y.S. Kim. "1997 Status of the Form Feature Recognition Method Using Convex Decomposition." *1997 ASME Design Engineering Technical Conferences* Sept. 1997: 14-17.
- [84] W.L. Wang, G.F. Jin, Y.B. Yan, and M.X. Wu. "Image Feature Extraction with the Optical Harr Wavelet Transform." *Optical Engineering* 34.4 (1995): 1238-1242.
- [85] K. Weiler. "Edge-based Data Structures for Solid Modeling in Curved-Surface Environments." *IEEE Computer Graphics and Applications* 5.1 (1985): 30-44.
- [86] J.R. Woodwark. "Some Specifications on Feature Recognition." *Computer-Aided Design* 20 (1988): 189-196.
- [87] M.C. Wu and C.R. Liu. "Analysis on Machined Feature Recognition Techniques Based on B-rep." *Computer-Aided Design* 28 (1996): 603-616.
- [88] X. Wu and B. Bhanu. "Gabor Wavelet Representation for 3D Object Recognition." *IEEE Transactions on Image Processing* 6.1 (1997): 47-64.
- [89] A.L. Yuille, David S. Cohen, and Peter W. Hallinan. "Feature Extraction from Faces Using Deformable Templates." *Proceedings of IEEE: Computer Vision and Pattern Recognition* June 1989: 104-109.
- [90] A.L. Yuille. "Zero Crossings on Lines of Curvatures." *Graphical Models and Image Processing* 45 (1989): 68-87.
- [91] I. Zeid. *CAD/CAM Theory and Practice*. New York: McGraw-Hill, 1991.
- [92] Z. Zhang. "Automated Image Inspection Using Wavelet Decomposition and Fuzzy Rule-Based Classifier." Ph.D. Dissertation, Department of Electrical and Computer Engineering, Iowa State University, 1995.