Wavelets and the Use of Curvature to Approximate Surfaces

by

Chang LI

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

Master of Science

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

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

WAVELETS AND THE USE OF CURVATURE TO APPROXIMATE SURFACES

BY

CHANG LI

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

MASTER OF SCIENCE

Acknowledgments

I thank my parents for their constant encouragement and support. Without their help I can not complete this dissertation. They really gave me the power to climb toward the top of a mountain.

I thank my supervisor Dr. Dereck Meek for instructing me in the field of curves and surfaces of computer graphics, for his patient interest and support throughout my Master's program, and for his help with impressive examples of curves and surfaces. His remarkable dedication to quality research is a strong inspiration. His geometric vision led me think more curves and surfaces simply and directly. I am also appreciative of examiners Dr. D.J. Walton and Dr. A. Gole for their careful reading the thesis.

I thank my relatives for their moral and financial support.

I thank my friends who gave me stimulation to finish this degree. I have benefited from their friendship and discussions in my years at the University of Manitoba.

Dedication

MY PARENTS SHUNHUA XIONG AND JIANLIN LI

Wavelets and the Use of Curvature to Approximate Surfaces

Abstract

by

Chang LI

Department of Computer Science

University of Manitoba

By using the wavelets and curvature, I tried to get a high quality compact representation of a surface. I get better results than simple Haar wavelets with curvature subdivision and Local Haar wavelets on the mathematical range data surface. To estimate the curvature of a curve represented by discrete data, a three point algorithm is developed. A normal approximation algorithm and an algorithm to estimate the Gaussian curvature are also developed for surface. The latter algorithm has a stable and fast convergence. To present background knowledge, I describe the multiresolution analysis with matrix and filter bank representation, the endpoint-interpolating B-spline wavelets, and basics of differential geometry. Several selection strategies for wavelets such as threshold and L^2 measurement are presented and tested. A simple location mapping algorithm for Haar wavelets is also studied. Finally I discuss the conclusions and future work.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Problems

Multiresolution analysis and wavelets have received considerable attention in recent year because they provide a useful and efficient tool for representing a function such as a surface at multiple levels of detail. Wavelets have been widely used in the field of computer graphics in areas such as multiresolution analysis of arbitrary topological type, compression of 3D model, graphics rendering, and acceleration of global illumination algorithms.

Initially, multiresolution analysis considers the representation of functions defined on the entire real line. The theory is then extended to functions on the unit interval. Therefore the theory can be applied to represent parametric curves, which are the most commonly used curves in computer graphics.

The extension of multiresolution analysis to surfaces on the unit square is straightforward by tensor product. It has been widely used on image compression and processing. Recent work has been done to extend multiresolution analysis to surfaces of arbitrary topological type.

In the case of multiresolution analysis of surfaces on the unit square, there are many problems that need to be solved. One problem is the following. Given a set of discrete points defined on a rectangular grid (range data), select coefficients, reconstruct an approximation surface to interpolate these points, and measure the quality of the approximation. This problem is related to image compression. In this dissertation, I will focus on solving this problem by using wavelets and curvature.

1.2 Dissertation Overview

This dissertation considers the problems of range data surface representation with wavelets, selection of coefficients, and surface reconstruction. With multiresolution analysis and wavelet of B-spline, some methods of surface representation, selection, and reconstruction are used. The estimation of curvature and applying it on a surface are the kernel of surface reconstruction. Chapter 2 presents some background knowledge on multiresolution analysis and B-spline curves and surfaces. Chapter 3 discusses the endpoint-interpolating B-spline wavelets for curves and surfaces. The standard and non-standard methods to produce wavelet representation of surfaces are explained.

Chapter 4 discusses the selection strategies to choose the wavelet coefficients. Several estimation methods of curvature for curves and surfaces are presented in Chapter 5. Chapter 6 discusses the subdivision and Local Haar Wavelet. Several comparisons to standard Haar wavelets are shown. Chapter 7 summarizes this work and suggests future work and applications. Appendix A presents a background knowledge of Differential Geometry about curvature.

1.3 Contributions

The contributions of this dissertation include:

- Defined a new non-standard method to construct two-dimensional wavelets.
- Tested three point circle method to estimate the curvature of a space curve on discrete points.
- Tested an algorithm to estimate the Gaussian curvature for range data surface by normal approximation.
- Presented subdivision by curvature method and applied it with wavelets.
- Developed Local Haar Wavelets (LHW), which are suitable for subdivision.

1.4 Notation

Notation used in this dissertation is represented in the following Table.

SYMBOL	DESCRIPTION
$\mathbb{R}$	The set of real numbers
x, j, etc	Scalar variables
$\mathbf{x}$, $\mathbf{p}$, etc.	Vectors, points
$\mathbf{I}$	The identity matrix
δ_{ij}	The Kronecker delta: 1 iff i=j, 0 otherwise
$\phi_{jk}(x)$	$\mathbb{R}^n \rightarrow \mathbb{R}$ The k-th scaling function at level j
$\Phi^j(x)$	$\mathbb{R}^n \rightarrow \mathbb{R}$ The matrix of $\phi_{jk}(x)$
V^j	The span of the scalar valued $\Phi^j(x)$
W^j	The orthogonal complement of V^j in V^{j+1}
$\Psi_{jk}(x)$	$\mathbb{R}^n \rightarrow \mathbb{R}$ The k-th wavelet at level j
$\Psi^j(x)$	$\mathbb{R}^n \rightarrow \mathbb{R}^l$ The matrix of $\Psi_{jk}(x)$ for a fixed j, where l is the length of $\Psi^j(x)$
$\langle f(x), g(x) \rangle$	A scalar giving the inner product of f(x) and g(x)

Chapter 2 Background

2.1 Terms

All the terms that are not well known are defined first before applying them in the following sections. The general terms of differential geometry are introduced in Appendix B.

2.2 Multiresolution Analysis

2.2.1 Basic Multiresolution Analysis

The theoretical development of multiresolution analysis is due to Mallat in 1989 [Mallat 89]. Dauberchies developed orthonormal wavelets and reappraised the research of wavelets [Daubechies 88]. Multiresolution analysis had become an abstract framework to introduce wavelets ([Stollnitz, DeRose, Salesin 94], [Chui 92], [Chui & Quak 92]). Multiresolution analysis and wavelets have found wide use in signal processing, image processing and compression ([Devore et al 92], [Averbuch et al 96], [Lewis & Knowles 92]), numerical analysis, and approximation theory. Recently multiresolution analysis and wavelets have been applied to solve problems in computer graphics, including curve modeling [Finkelstein & Salesin 94], modeling surfaces of arbitrary topological type ([Lounsbery 94] and [Eck et al 95]), radiosity [Schroder et al. 93], and optimization for surface interpolation [Pentland 92].

The basic concept behind multiresolution analysis is to decompose a complicated function into a "simpler" low-resolution part together with a set of details called wavelet coefficients. By omitting small coefficients we can efficiently construct an

approximation of a function. Multiresolution analysis actually provides a flexible tool to handle a function in different levels of detail. It can be applied to various spaces such as real line, mesh, and vectors. In the next sections I'll focus on applying multiresolution analysis to a vector space.

2.2.2 Multiresolution Analysis of a Vector Space

The multiresolution analysis of a vector space includes two basic elements:

1. A nested set of vector spaces

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

V^j contains functions of resolution j and as j increases the resolution of V^j increases. The *scaling functions* refer to the basis functions for V^j.

2. For an arbitrary pair of functions f and g in V^j define an inner product <f, g>. The orthogonal complement space W^j of V^j in V^{j+1} is defined as the set of functions in V^{j+1} such that their inner product with any one of the functions in V^j is zero. That is any function in V^{j+1} can be written uniquely as an orthogonal decomposition of a function in V^j and a function in W^j. It is often written as $V^{j+1} = V^j \oplus W^j$. W^j is known as the wavelet space, and the basis functions of W^j are called *wavelets*. Figure 2.1 shows this.

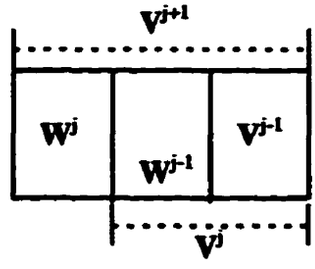


Figure 2.1 Vector and wavelet space

3. Generally the nested spaces V^j are generated by *translations* and *dilations* of a single refinable function $\phi(x)$. Over the real line this is represented by the refinement equation

$$\phi(x) = \Sigma_i \, p_i \phi(2x-i) \quad ,$$

where p_i is constant. The constant factor 2 acts as a *dilation*, and the negative i is considered to be a *translation*. *Refinability* for multiresolution analysis means that a coarse-level scaling function can be expressed as a linear combination of fine-level scaling functions. For example the B-spline basis functions introduced later are refinable.

2.2.3 Matrix Representation

Let $\phi_{ji}(x)$ denote the i^{th} scaling function on the vector space V^j, and $\psi_{ji}(x)$ denote the i^{th} wavelet basis function on the vector space W^j. x is the parameter of the functions. It is convenient to represent the scaling functions as

$$\Phi^j(x) = (\phi_{j0}(x), \phi_{j1}(x), \ldots, \phi_{jM_{j-1}}(x))$$

where M_j is the dimension of V^j.

For wavelets let

$$\Psi^j(x) = (\psi_{j0}(x), \psi_{j1}(x), \ldots, \psi_{jN_{j-1}}(x)) \, ,$$

where N_j is the dimension of W^j. Note that $N_j = M_{j+1} - M_j$.

Since the subspaces V^j are nested and the scaling functions are required to be refineable, there exists a constant matrix P^j for each j = 1, 2, ... such that

$$\Phi^{j-1}(x) = \Phi^j(x) * P^j \qquad (2.1)$$

the P^j is a $M_j * M_{j-1}$ matrix . That is scaling functions at level j-1 can be represented as a linear combination of scaling functions at level j.

Because wavelet space W^{j-1} is a subspace of V^j and the basis functions are refinable, there is also a constant matrix Q^j such that

$$\Psi^{j-1}(x) = \Phi^j(x) * Q^j \qquad (2.2)$$

the Q^j is an $M_j * (M_j - M_{j-1})$ matrix. That is each wavelet at level j-1 is also represented as a linear combination of scaling functions at level j.

Combining (2.1) and (2.2) we have

$$(\Phi^{j-1}(x) \ \ \Psi^{j-1}(x)) = \Phi^j(x) * (P^j \ \ Q^j) \qquad (2.3)$$

where $(P^j \ Q^j)$ is a $M_j * M_j$ matrix.

Example:

For Haar basis (refer to 3.1.1) given a level j there are 2^j scaling functions and wavelets. In the V^2,

$$P^2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad Q^2 = \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}$$

Suppose

$$\Phi^1(x) = [\phi_{10}(x), \phi_{11}(x)], \Psi^1(x) = [\psi_{10}(x), \psi_{11}(x)]$$

$$\Phi^2(x) = [\phi_{20}(x), \phi_{21}(x), \phi_{22}(x), \phi_{23}(x)]$$

then

$$[\Phi^1(x)\ \ \Psi^1(x)] = \Phi^2(x)\begin{pmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$

2.2.4 Filter Bank

Suppose a function $f_j(x)$ in V^j is expressed by the scaling functions as

$$f_j(x) = \sum_{i=0}^{M_j-1} c_{ji}\phi_{ji}(x)$$

where x is the variable, and the c_{ji} are the coefficients of the linear combination.

Let $C^j = (c_{j0}, c_{j1}, ..., c_{jMj-1})^T$ be the column vector of the coefficients. Using *linear filtering* (linear combination) and *downsampling* (simply drop off the coefficients) on the M_j entries of C^j, we can create a low-resolution C^{j-1} with a smaller number of coefficients M_{j-1}. It can be written as

$$C^{j-1} = A^j * C^j \qquad (2.4)$$

where A^j is a constant $M_{j-1} * M_j$ matrix.

The detail lost by downsampling can be captured as another column vector D^{j-1} where

$$D^{j-1} = B^j * C^j \qquad (2.5)$$

and B^j is a constant $(M_j - M_{j-1}) * M_j$ matrix.

The matrices A^j and B^j are called *analysis filters*. The above splitting process is called *analysis*. With A^j and B^j chosen such that $(P^j \;\; Q^j)\begin{pmatrix} A^j \\ B^j \end{pmatrix} = I$, the C^j can be recovered from C^{j-1} and D^{j-1} with

$$C^j = P^j * C^{j-1} + Q^j * D^{j-1} \qquad (2.6)$$

or equivalently with

$$C^j = (P^j \; Q^j)\begin{pmatrix} C^{j-1} \\ D^{j-1} \end{pmatrix} \qquad (2.7)$$

This procedure is called *synthesis*. P^j and Q^j are called *synthesis filters*.

The procedure for splitting C^j into C^{j-1} and D^{j-1} can be recursively applied to C^{j-1}. This is called *analysis filter bank*. The reverse procedure is called *synthesis filter bank*. See Figure 2.2 and Figure 2.3.

analysis filter bank

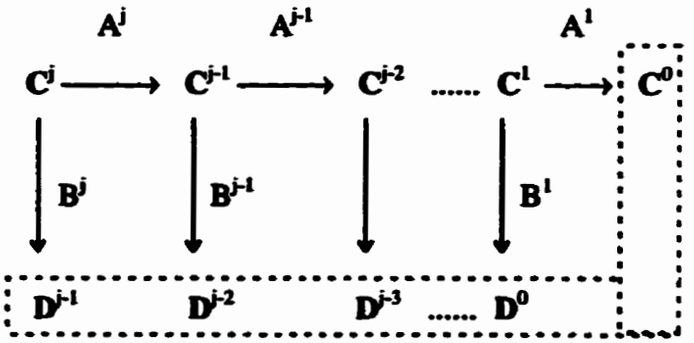


Figure 2.2 Analysis filter bank

synthesis filter bank

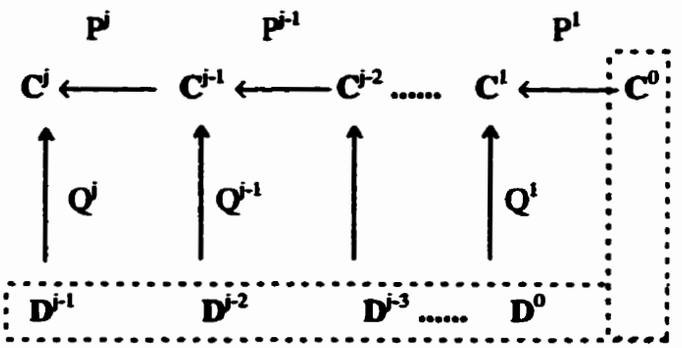


Figure 2.3 Synthesis filter bank

C^0, D^0, D^1, ..., D^{j-1} is the wavelet transform sequence. The relationship between analysis and synthesis matrices can be written

$$(P^j Q^j)^{-1} = \begin{pmatrix} A^j \\ B^j \end{pmatrix} \qquad (2.8)$$

Once we have chosen scaling functions and their *synthesis filters* P^j, the synthesis filters Q^j are constrained. Since we require all the functions $\phi_{j-1,k}(x)$ in $\Phi^{j-1}(x)$ and

$\psi_{j-1,l}(x)$ in $\Psi^{j-1}(x)$ satisfy the inner product $<\phi_{j-1,k}(x), \psi_{j-1,l}(x)> = 0$ for all k and l. It is convenient to represent it in the matrix form,

$$[\,<\Phi^{j-1}(x), \Psi^{j-1}(x)>] = \mathbf{0}.$$

From (2.2) $\Psi^{j-1}(x) = \Phi^{j}(x)*Q^{j}$, we have $[<\Phi^{j-1}(x), \Psi^{j-1}(x)>] = [<\Phi^{j-1}(x),\Phi^{j}(x)>]*Q^{j} = \mathbf{0}$. The columns of Q^{j} must form a basis for the null space of $[<\Phi^{j-1}(x), \Psi^{j-1}(x)>]$. Refer to [Golub & Van Loan 89] for discussion about this condition of null space. Because there are many bases for the null space of a matrix, there are many wavelet bases for a given space W^{i}. Therefore there are more constraints required to uniquely figure out the Q^{j} matrices other than the condition orthgonality.

For example we could require the columns of Q^{j} to be sparse and with a minimal number of consecutive non-zeros. See [Finkelstein & Salesin 94] for their construction of cubic spline wavelets. Refer to [Press et al 92] for numerical algorithms.

2.3 B-spline Curves

2.3.1 Definition

The basic B-spline is described in the books [de Boor 72], [Farin 93], [Farin 95], and [Piegl & Tiller 87]. Many authors studied the other kinds of splines. [Meek & Walton 92] presented a method to approximate a group of discrete data by G^1 arc splines. [Forsey & Bartels 95] described surface fitting with hierarchical splines.

A B-spline curve P(x) of degree d is defined by parameter x and control vertices V_i as

$$\mathbf{P}(x) = \sum_{i=0}^{n} N_i^d(x) * \mathbf{V}_i$$

Where $x_{min} \le x \le x_{max}$ and $1 \le d \le n$. The V_i are n+1 control vertices and $N_i^d(x)$ are the normalized B-spline basis functions. d is the degree of the B-spline basis function.

Define a sequence $[x_0, x_1, \ldots\ldots, x_{n+d+1}]$ called *knots*. It satisfies the relation $x_i \le x_{i+1}$ where $0 \le i \le n+d$. The parameter x is in the interval $[x_d , x_{n+1}]$. For the i^{th} normalized B-spline basis function of degree r, the basis function $N_i^r(x)$ are defined by recursion formulas:

$$N_i^0(x) = \begin{cases} 1 & \text{if } x_i \le x < x_{i+1} \\ 0 & \text{otherwise} \end{cases} \qquad (2.10a)$$

$$N_i^r(x) = \frac{x - x_i}{x_{i+r} - x_i} N_i^{r-1}(x) + \frac{x_{i+r+1} - x}{x_{i+r+1} - x_{i+1}} N_{i+1}^{r-1}(x) \qquad (2.10b)$$

$$i = 0, ..., n \quad r = 1, ..., d$$

Where $d > 0$ is the maximum degree of the B-spline basis function. If either $N_i^{r-1}(x)$ *or* $N_{i+1}^{r-1}(x)$ is zero, the associated term in (2.10b) is omitted. Refer to the book [Rogers & Adams 90].

Based upon the knot sequence, we can classify the B-spline to be ***uniform, open uniform, and nonuniform.***

1) In a ***uniform knot sequence***, the knot values are evenly spaced, that is $x_{i+1} = x_i +$ c where c is a constant. For example, the knot sequence [0, 1/4, 2/4, 3/4] is uniform with c=1/4. The uniform knot sequence yields periodic uniform B-spline basis functions and each basis function is a translation of the other.

2) An ***open uniform knot sequence*** has d+1 equal knot values defined at the endpoints and the interior knots are uniform. Here d is the degree of B-spline basis function. For example, in the case of cubic d = 3, the sequence [0, 0, 0, 0, 1/2, 1, 1, 1, 1] is open uniform.

If the number of control vertices n+1 is equal to d+1 with an open uniform knot sequence, then the B-spline reduces to a Bezier curve, and the B-spline basis functions reduce to the Bernstein basis functions. The Bernstein basis functions are

$$B_i^d(x) = \binom{d}{i} x^i (1-x)^{d-i} \quad x \in [0, 1]$$

where d is the degree of the Bernstein basis function.

The knot sequence is just d+1 zeros followed by d+1 ones. For example, the cubic Bezier curve has the knot sequence [0, 0, 0, 0, 1, 1, 1, 1].

3) ***Non-uniform knot*** sequence has no limitations except that it is a non-decreasing sequence.

The ***endpoint-interpolating B-spline*** of degree d in [0, 1] is an open uniform B-spline. The $N_0^d(x),\ldots,N_n^d(x)$ are the B-spline basis of degree d with d-1 continuous derivatives. Its interior knots are $x_{d+1}, \ldots, x_n$, the endpoint knots are $x_0 = x_1 = \ldots = x_d = 0$, and $x_{n+1} = x_{n+2} = \ldots = x_{n+d+1} = 1$.

To make the endpoint-interpolating B-spline refinable, we set n to be 2^j +d-1 and $x_d, \ldots, x_{n+1}$ produce 2^j equally-spaced interior intervals. Therefore there are 2^j+d B-spline basis functions used to define the endpoint-interpolating B-spline scaling functions for wavelets. Figure 2.4 shows the knots values. For cubic endpoint-interpolating B-spline with d=3 and level j=2, the knot sequence is [0 0 0 0 1/4 2/4 3/4 1 1 1 1].

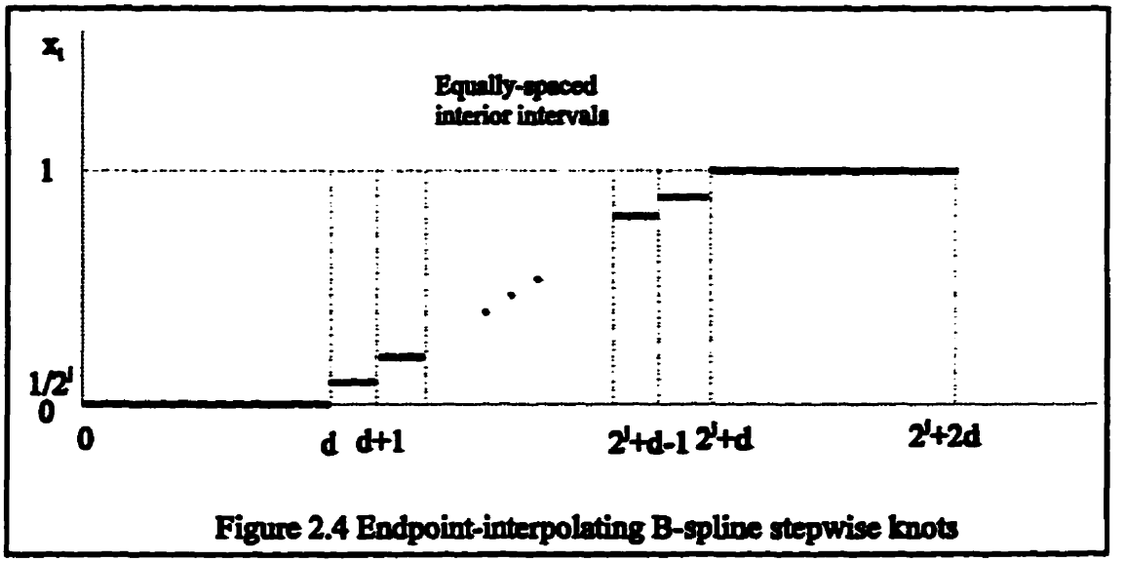


Figure 2.4 Endpoint-interpolating B-spline stepwise knots

2.3.2 Properties

The B-spline curve and basis functions have some important properties[Rogers & Adams 90] :

- For any given parametric x, each B-spline basis function is non-negative and the sum of the B-spline basis functions is 1, that is

$$\sum_{i=0}^{n} N_i^r(x) \equiv 1 \qquad \text{and } N_i^r(x) \geq 0, \text{ for r} = 0,1,...,d$$

- The B-spline curve has the *variation diminishing* property, that is it does not cross any straight line more often than the straight line crosses its defining control polygon. The maximum degree of the B-spline curve is one less than the number of control vertices. For example, the largest degree B-spline with 4 control vertices is cubic.

- The B-spline curve is bounded by the convex hull of its defining control polygon. If the control vertices are collinear, then the B-spline is a straight line.

- The B-spline of degree d is continuous and its first, second, ..., and (d-1)th derivatives are all continuous (assuming no multiple knots and no multiple control points).

2.3.3 Derivatives

1) First Derivatives [Rogers & Adams 90, Page 337]

From equation (2.10b), we have

$$N_i^r{}'(x) = \frac{N_i^{r-1}(x) + (x - x_i)N_i^{r-1}{}'(x)}{x_{i+r} - x_i} + \frac{(x_{i+r+1} - x)N_{i+1}^{r-1}{}'(x) - N_{i+1}^{r-1}(x)}{x_{i+r+1} - x_{i+1}} \qquad (2.11)$$

For all x, $N_i^0{}'(x) = 0$. When r = 1 in (2.11) and using (2.10), we have

$$N_i^1{}'(x) = \frac{N_i^0(x)}{x_{i+1} - x_i} - \frac{N_{i+1}^0(x)}{x_{i+2} - x_{i+1}} = \begin{cases} \dfrac{1}{x_{i+1} - x_i} & x \in [x_i, x_{i+1}) \\ \dfrac{-1}{x_{i+2} - x_{i+1}} & x \in [x_{i+1}, x_{i+2}) \\ 0 & \textit{otherwise} \end{cases} \qquad (2.12)$$

Example:

For uniform knot sequence, x_{i+1}-x_i = x_{i+2}-x_{i+1} = c where c is a constant, so

$$N_i^{1\,\prime}(x) = \begin{cases} 1/c & x \in [x_i, x_{i+1}) \\ -1/c & x \in [x_{i+1}, x_{i+2}) \\ 0 & \textit{otherwise} \end{cases}.$$

2) Second Derivatives

$$N_i^{r\,\prime\prime}(x) = \frac{2N_i^{r-1\,\prime}(x) + (x - x_i)N_i^{r-1\,\prime\prime}(x)}{x_{i+r} - x_i} + \frac{(x_{i+r+1} - x)N_{i+1}^{r-1\,\prime\prime}(x) - 2N_{i+1}^{r-1\,\prime}(x)}{x_{i+r+1} - x_{i+1}} \quad (2.13)$$

For all x $N_i^{0\,\prime\prime}(x) = N_i^{1\,\prime\prime}(x) = 0$. When r = 2 in (2.13) and using (2.12) we have

$$N_i^{2\,\prime\prime}(x) = 2\left(\frac{N_i^{1\,\prime}(x)}{x_{i+2} - x_i} - \frac{N_{i+1}^{1\,\prime}(x)}{x_{i+3} - x_{i+1}}\right) = \begin{cases} \dfrac{2}{(x_{i+2} - x_i)(x_{i+1} - x_i)} & x \in [x_i, x_{i+1}) \\ \dfrac{-2}{x_{i+2} - x_{i+1}}\left[\dfrac{1}{x_{i+2} - x_i} + \dfrac{1}{x_{i+3} - x_{i+1}}\right] & x \in [x_{i+1}, x_{i+2}) \\ \dfrac{2}{(x_{i+3} - x_{i+1})(x_{i+3} - x_{i+2})} & x \in [x_{i+2}, x_{i+3}) \\ 0 & \textit{otherwise} \end{cases}$$

Example:

For uniform knot sequence we have

$$N_i^{2\,\prime\prime}(x) = \begin{cases} 1/c^2 & x \in [x_i, x_{i+1}) \\ -2/c^2 & x \in [x_{i+1}, x_{i+2}) \\ 1/c^2 & x \in [x_{i+2}, x_{i+3}) \\ 0 & \textit{otherwise} \end{cases}$$

2.4 B-spline Surfaces

2.4.1 Definition

The natural definition of B-spline surface is by tensor-product

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} V_{ij} N_i^r(u) N_j^s(v) \quad ,$$

where $N_i^r(u)$ *and* $N_j^s(v)$ are the B-spline basis functions in the u and v directions respectively (refer to (2.10)); x_i and y_j are the knot sequences in the u and v directions; V_{ij} is the control vertices of a defining polyhedral net; m and n are the number of control vertices in the u and v directions respectively. The two knot sequences can be different.

2.4.2 Properties

- The parametric continuity of the surface in each parametric direction depends on the degree in each direction.
- The variation diminishing property for a B-spline surface is currently not known.
- The surface lies within the convex hull of the defining polyhedral net.
- The surface is invariant under an affine transformation. That is you can transform the polyhedral net to transform the surface.

2.4.3 Derivatives

The partial derivatives of B-spline surface are defined as

$$S_u(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} V_{ij} N_i^{r\prime}(u) M_j^s(v)$$

$$S_v(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} V_{ij} N_i^{r}(u) M_j^{s\prime}(v)$$

$$S_{uv}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} V_{ij} N_i^{r\prime}(u) M_j^{s\prime}(v)$$

$$S_{uu}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} V_{ij} N_i^{r\prime\prime}(u) M_j^{s}(v)$$

$$S_{vv}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} V_{ij} N_i^{r}(u) M_j^{s\prime\prime}(v)$$

Chapter 3 Endpoint-interpolating B-spline Wavelets

Let $V^j(d)$ be the space spanned by the endpoint-interpolating B-spline scaling functions of degree d with 2^j uniform intervals, then $V^0(d) \subset V^1(d) \subset V^2(d)$ ..... The columns of the synthesis matrix P^j are sparse, reflecting the fact that the B-spline basis functions are locally supported. The first and last d columns are relatively complicated because of the end point condition of B-spline, but the interior columns are shifted versions of the column d+1. The elements of the interior columns are given by binomial coefficients. Refer to Chapter 2, [Stollnitz, et al, 1994], and [Quak & Weyrich 93].

Figures 3.1 and 3.2 illustrate how each vector $V^0(0)$ can be represented by the basis functions of $V^1(0)$. The key is that the box function $\phi_{00}(x)$ can be expressed by the linear combination of two box functions $\phi_{10}(x)$ and $\phi_{11}(x)$. In this way we can see $V^0(0)$ is a subspace of $V^1(0)$.

The inner product is defined as

$$\boxed{< f, g > = \int_0^1 f(x)g(x)dx}$$

B-spline wavelets are determined by the matrix equation, $[<\Phi^{j+1}(x), \Phi^j(x)>]Q^j = 0$ (refer to (2.9)). In the approach taken by [Finkelstein & Salesin 1994] and [Stollnitz et al. 1994] to construct the matrix Q^j, the columns of Q^{j+1} are required to be sparse and have a minimal number of consecutive non-zeros. [Chui & Quak 1992] use the derivative and interpolation properties of B-splines to determine the Q^j.

$\mathbf{C}^{j-1}$ and $\mathbf{D}^{j-1}$ can be computed from $\mathbf{C}^j$ by solving the sparse linear system $(\mathbf{P}^j\ \mathbf{Q}^j)\begin{pmatrix}\mathbf{C}^{j-1}\\ \mathbf{D}^{j-1}\end{pmatrix}=\mathbf{C}^j$. Since $(\mathbf{P}^j\ \mathbf{Q}^j)$ is a sparse matrix, [Stollnitz, et al, 1994] reported a linear time solution to above equation with LU decomposition.

3.1 One-dimensional B-spline Wavelets

3.1.1 Haar Wavelets

When degree of B-spline wavelet basis function is zero the B-spline wavelet basis becomes the Haar wavelet basis. Haar wavelet basis is the simplest wavelet basis. In the case of one-dimensional wavelets, we consider that the domain of the basis functions is the interval [0,1). We divide the interval equally into 2^j pieces in which j is called the *level*. For each j there exists a vector space V^j which includes all the 2^j piecewise-constant functions on the interval [0, 1). V^{j+1} is produced from V^j by bisecting each interval in V^j. Every vector in V^j is also contained in V^{j+1}, because each constant function in V^j can be represented as the linear combination of two constant functions in V^{j+1}. Thus the spaces V^j are nested. For each vector space V^j, the Haar scaling functions are defined as the set of dilation and translation of the "box" function $\phi(x)$.

$$\phi_{ji}(x) = \phi\,(2^j x - i) \quad , \qquad (3.1)$$

where $i = 0, 1, ..., 2^j-1$ and

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

$[\phi_{j0}(x), \phi_{j1}(x), ..., \phi_{j2^j-1}(x)]$ are Haar scaling functions. It is the set of functions spanning the vector space V^j. In a similar fashion, the wavelet $\psi_{ji}(x)$ is defined to be the functions that span the orthogonal complement space W^j. Under the chosen inner product, every wavelet $\psi_{ji}(x)$ of W^j is orthogonal to every scaling basis function of V^j.

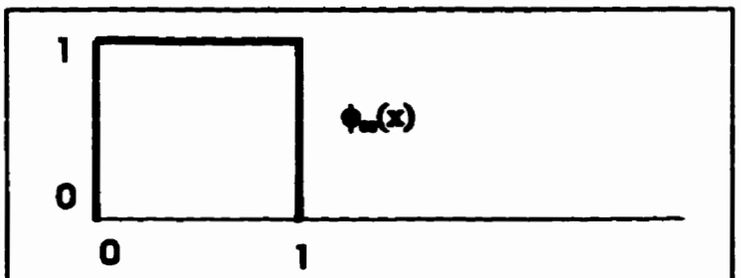


Figure 3.1 The Haar scaling function for V^0 (j = 0)

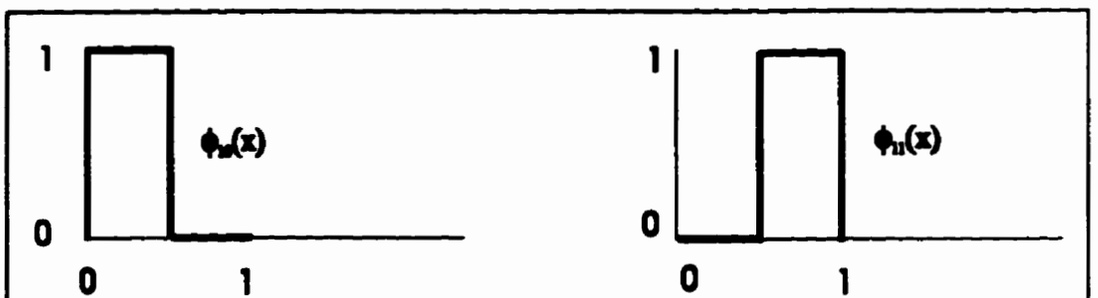


Figure 3.2 The Haar scaling functions for V^1 (j = 1)

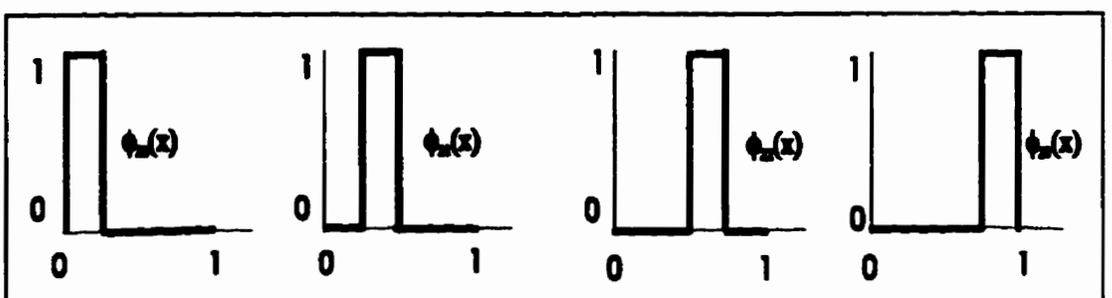


Figure 3.3 The Haar scaling functions for V^2 (j = 2)

Chapter 3 Endpoint-interpolating B-spline Wavelets

The Haar wavelets are

$$\psi_{ji}(x) = \psi\,(2^j x - i)$$

where $i = 0,1, \ldots\ldots, 2^j - 1$

$$\Psi(x) = \begin{cases} 1 & 0 \le x < 1/2 \\ -1 & 1/2 \le x < 1 \\ 0 & \textit{otherwise} \end{cases}$$

Normalized basis functions have a norm 1. That is $\langle f, f \rangle = 1$ where f is a basis function. The normalized Haar scaling functions and wavelets are

$$\phi_{ji}(x) = 2^{j/2}\phi(2^j x - i)$$
$$\psi_{ji}(x) = 2^{j/2}\psi(2^j x - i)$$

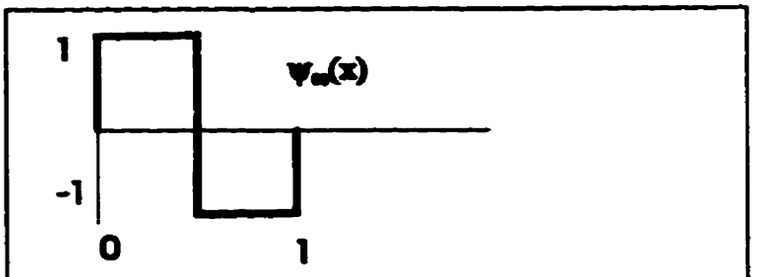


Figure 3.4 The Haar wavelet function for W^0 (j = 0)

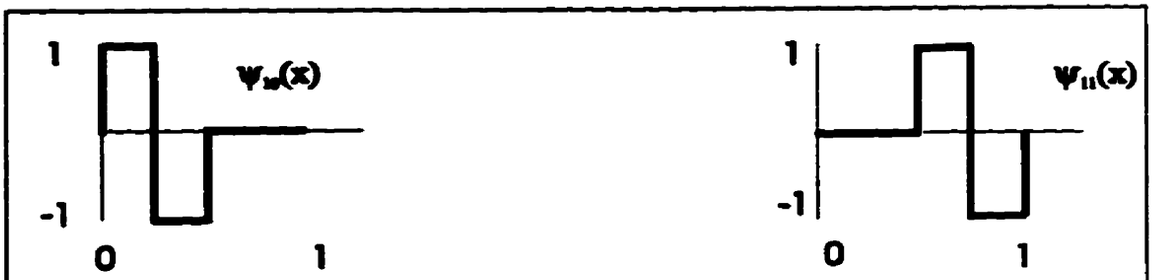


Figure 3.5 The Haar wavelets function for W^1 (j = 1)

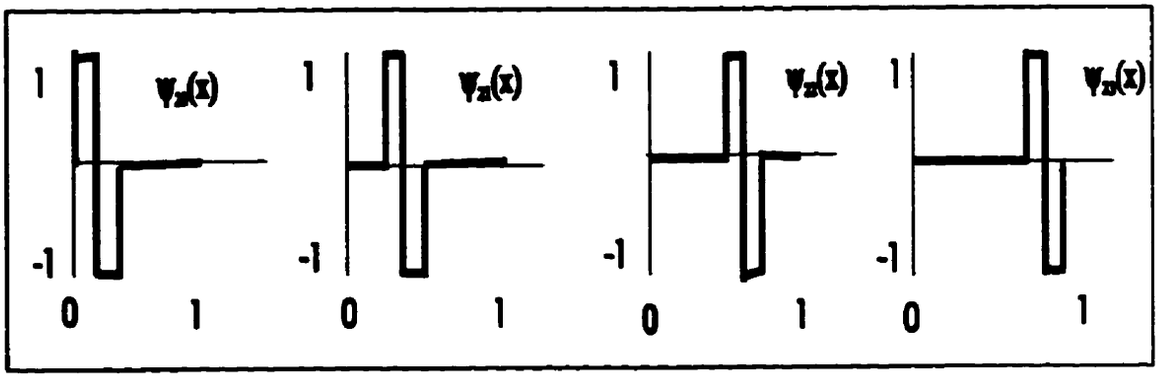


Figure 3.6 Haar wavelet functions for W^j (j = 2)

For multiresolution analysis and matrix representation, the synthesis matrices P^j and Q^j for Haar basis are

$$P^j = \begin{pmatrix} 1 & 0 & & & \\ 1 & 0 & & & \\ 0 & 1 & & & \\ 0 & 1 & & & \\ & & \cdot & & \\ & & & 0 & 1 \\ & & & 0 & 1 \end{pmatrix}, \qquad Q^j = \begin{pmatrix} 1 & 0 & & & \\ -1 & 0 & & & \\ 0 & 1 & & & \\ 0 & -1 & & & \\ & & \cdot & & \\ & & & 0 & 1 \\ & & & 0 & -1 \end{pmatrix}$$

The Haar basis functions are simple and orthogonal, have very compact support, and are non-overlapping. Compact support means the wavelets only relate to the local neighbor points, so the synthesis matrices are sparse. This is an important advantage in real applications.

However the Haar basis is poor in the applications of computer graphics because of its lack of continuity.

3.1.2. Linear Endpoint-Interpolating B-spline Wavelets

If we choose the degree d of the B-spline basis functions to be 1, the scaling functions are $N_0^1(x), N_1^1(x), \ldots\ldots, N_n^1(x)$. These wavelets are continuous, or are C^0. The following diagrams are from [Stollnitz et al 94]

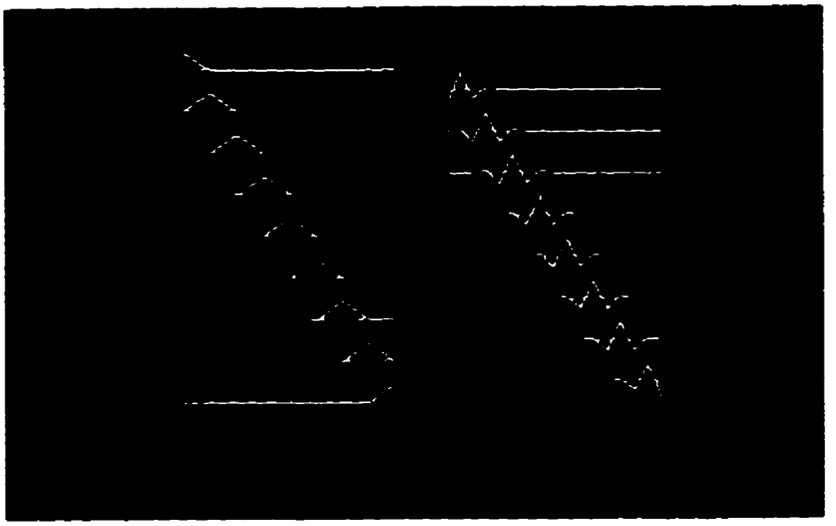


Figure 3.7 Linear endpoint-interpolating B-spline scaling functions and wavelets for $n = 2^3$

3.1.3 Quadratic Endpoint-Interpolating B-spline Wavelets

If we choose the degree d of the B-spline basis functions to be 2, their first derivatives are continuous so the wavelets are C^1

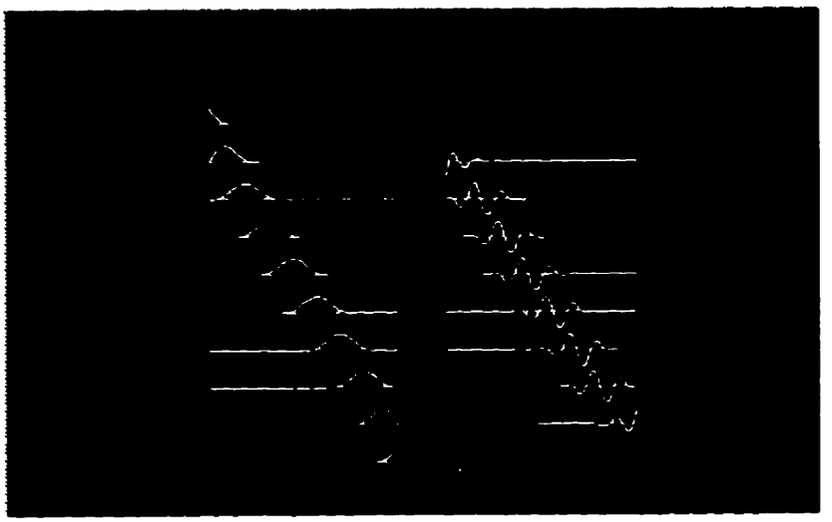


Figure 3.8 Quadratic endpoint-interpolating B-spline scaling functions and wavelets for $n = 2^3$

3.1.4. Cubic Endpoint-Interpolating B-spline Wavelets

If we choose the degree of cubic B-spline wavelet basis functions to be 3, their second derivatives are continuous so the wavelets are C^2. When we compute the curvature of a curve and a surface, second derivatives are required, and the cubic B-spline is suitable for this purpose.

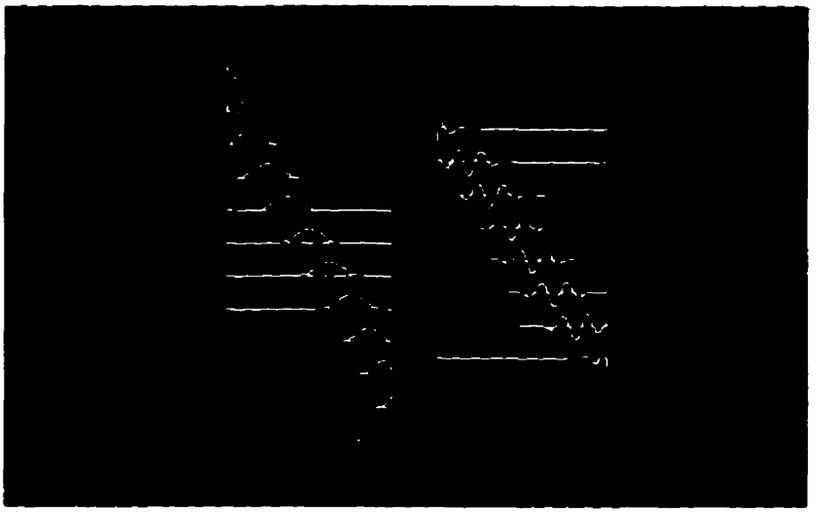


Figure 3.9 Cubic endpoint-interpolating B-spline scaling functions and wavelets for $n = 2^3$

3.2 Two-dimensional B-spline Wavelets

In the following sections, I assume that two-dimensional image is defined on a rectangular domain. The rectangle topology makes it easy to construct a two-dimensional wavelet from one-dimensional wavelets. There are several ways to do so. We will discuss the standard tensor product wavelets and non-standard wavelets in the next sections.

3.2.1 Standard Tensor Product Wavelets

Intuitively, the tensor product construction is simply the application of an one-dimensional transformation to all the rows and then to all the columns in the rectangle domain. In the wavelet transformation, the row wavelet transformation sets the wavelet coefficients with more detail from left to right. And then the column wavelet transformation treats these row wavelet coefficients as input and applies to every row from top to bottom. The final two-dimensional wavelet coefficients are arranged in the way that the top-left corner has the coarse coefficients and the fine coefficients are at the bottom-right.

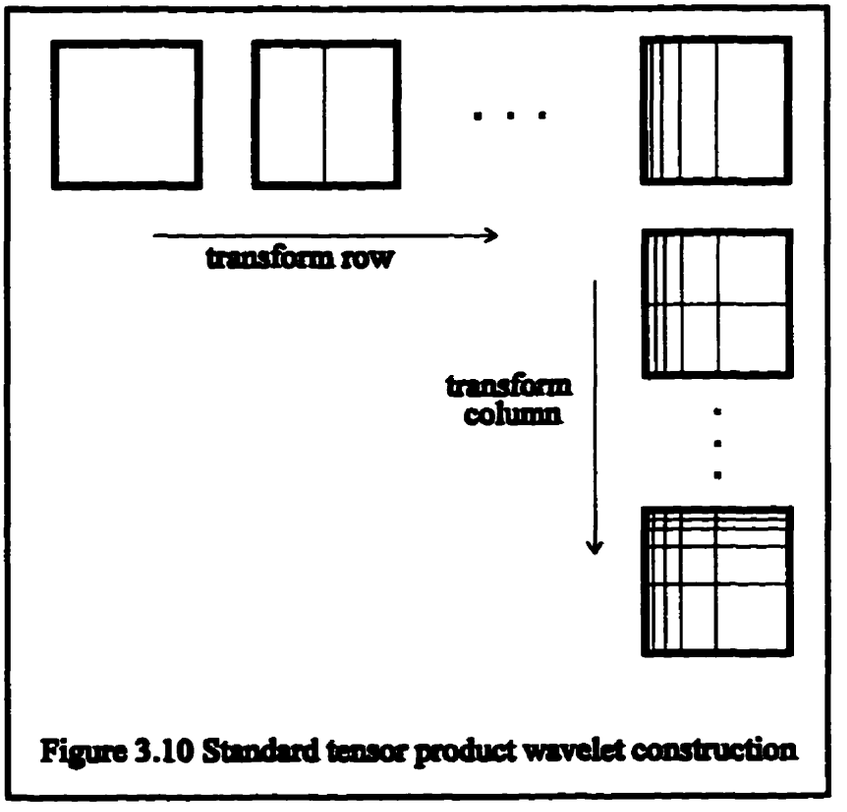


Figure 3.10 Standard tensor product wavelet construction

3.2.2 Non-standard Wavelets

To get the non-standard wavelet transformation, first apply one step of a one-dimensional wavelet transformation to each row and apply one step of wavelet transformation to each column. Repeat above procedure on the top-left quarter area. Repeat the above procedure until you reach the minimum area. See Figure 3.11

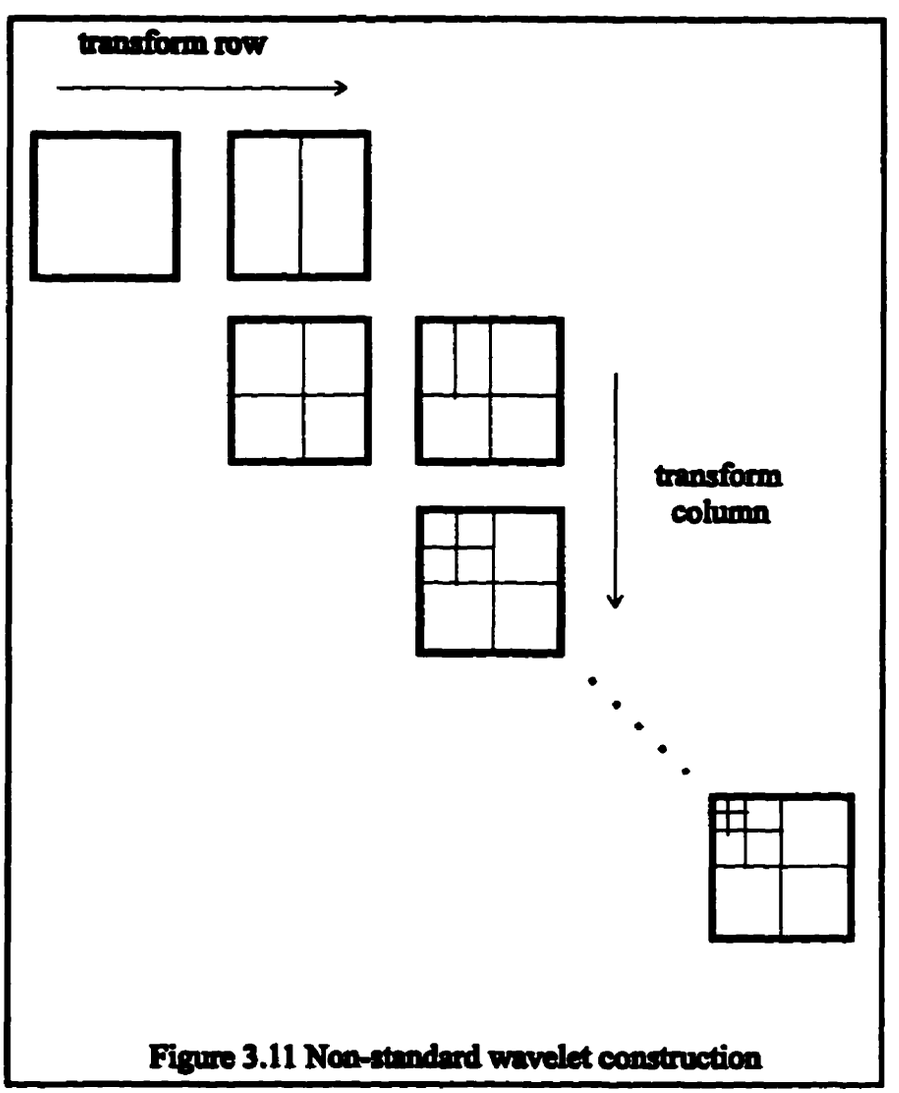


Figure 3.11 Non-standard wavelet construction

Since the procedure above only applies the transformation to the coarse area recursively, we may have another reasonable construction that is called Non-standard II. It transforms rows to *coarse row area* and transforms columns to *coarse column area* recursively. This is similar to the tensor product transformation., but it applies the wavelet transformation in alternation of row and column. See Figure 3.12

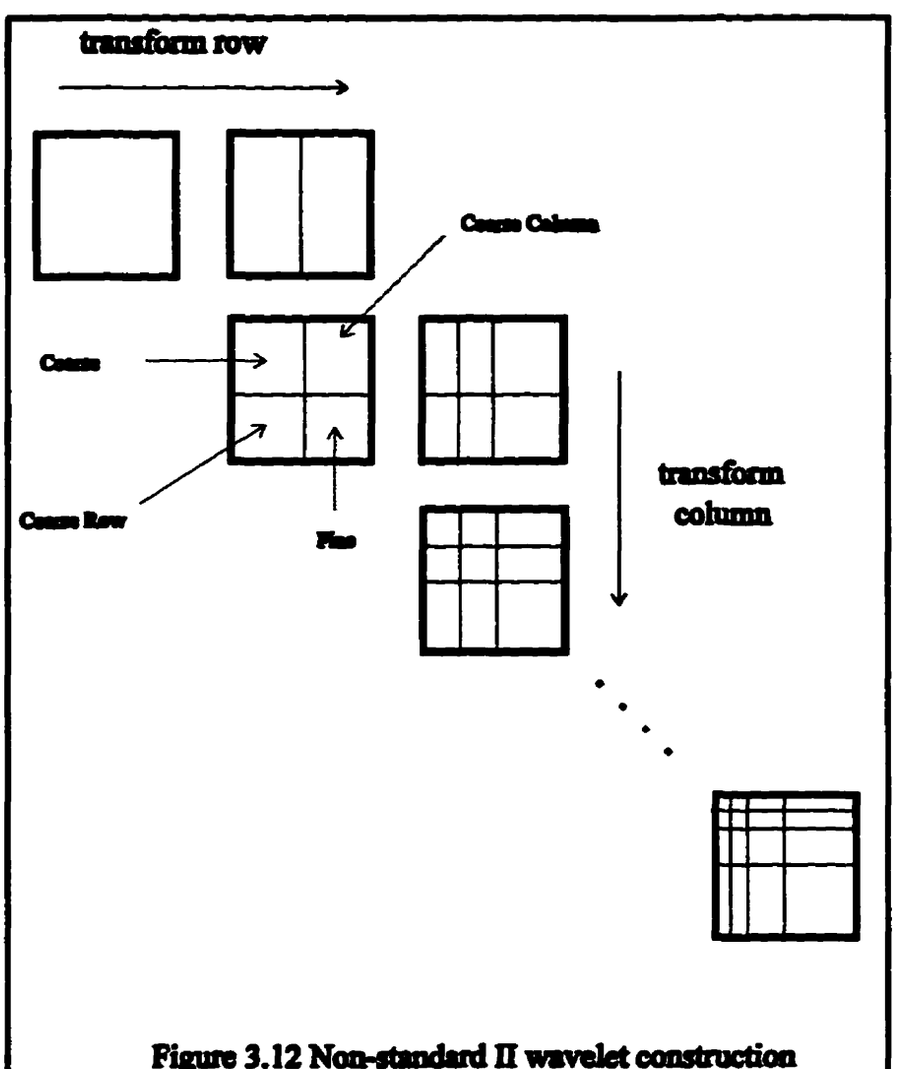


Figure 3.12 Non-standard II wavelet construction

3.2.3 Two-dimensional B-spline Wavelet Basis Functions

Suppose the B-spline basis functions along the x direction are $\Phi(x) = [\phi_{00}(x), \psi_{00}(x), \ldots\ldots, \psi_{mm}(x)]$ $m = 2^j - 1 + d$ and the functions along the y direction are $\Phi(y) = [\phi_{00}(y), \psi_{00}(y), \ldots\ldots, \psi_{nn}(y)]$ and $n = 2^k - 1 + d$. The basis functions of tensor product are elements of the matrix $\Phi^T(x)\Phi(y)$. The $m = n = 1$ and $d = 0$ basis functions are represented as a matrix of the product,

$$\begin{pmatrix} \phi_{00}(x) \\ \psi_{00}(x) \\ \psi_{10}(x) \\ \psi_{11}(x) \end{pmatrix} \begin{pmatrix} \phi_{00}(y) & \psi_{00}(y) & \psi_{10}(y) & \psi_{11}(y) \end{pmatrix}$$

The non-standard wavelet basis functions are produced by first defining a two-dimensional coarse scaling function

$$\boxed{\phi\phi(x,y) = \phi(x)\phi(y)}$$

and three wavelet functions

$$\boxed{\begin{aligned} \phi\psi(x,y) &= \phi(x)\psi(y) \\ \psi\phi(x,y) &= \psi(x)\phi(y) \\ \psi\psi(x,y) &= \psi(x)\psi(y) \end{aligned}}$$

They can be represented in the form of a recursive tensor product for the Haar wavelets. For example, for $m = n = 1$ and $d = 0$, let $U(x) = \begin{pmatrix} U_1(x) \\ U_2(x) \end{pmatrix}$ where $U_1(x) = (\phi_{00}(x), \psi_{00}(x))^T$ and $U_2(x) = (\psi_{10}(x), \psi_{11}(x))^T$; $V(y) = (V_1(y)\ V_2(y))$ where $V_1(y) = (\phi_{00}(y), \psi_{00}(y))$ and $V_2(y) = (\psi_{10}(y), \psi_{11}(y))$.

The non-standard basis function is defined by Θ as $U(x) \Theta V(y) = U(x)V(y)$ if U or V has one element and

$$U(x)\Theta V(y) = \begin{pmatrix} U_1(x)\Theta V_1(y) & U_1(x)V_2(y) \\ U_2(x)V_1(y) & U_2(x)V_2(y) \end{pmatrix}$$

In above example

$$U(x)\Theta V(y) = \begin{pmatrix} \begin{pmatrix} \phi_{00}(x) \\ \psi_{00}(x) \end{pmatrix} \Theta \left(\phi_{00}(y)\ \psi_{00}(y)\right) & \begin{pmatrix} \phi_{00}(x) \\ \psi_{00}(x) \end{pmatrix} \left(\psi_{10}(y)\ \psi_{11}(y)\right) \\ \begin{pmatrix} \psi_{10}(x) \\ \psi_{11}(x) \end{pmatrix} \left(\phi_{00}(y)\ \psi_{00}(y)\right) & \begin{pmatrix} \psi_{10}(x) \\ \psi_{11}(x) \end{pmatrix} \left(\psi_{10}(y)\ \psi_{11}(y)\right) \end{pmatrix}$$

Each of construction has its own advantages. We define the support of a wavelet as the domain where the wavelet is non-zero. The standard tensor product wavelet is simple, but has non-square support. The non-standard wavelets have square supports.

Chapter 4. Selection Strategies

An application of wavelets for data compression typically requires three distinct stages:

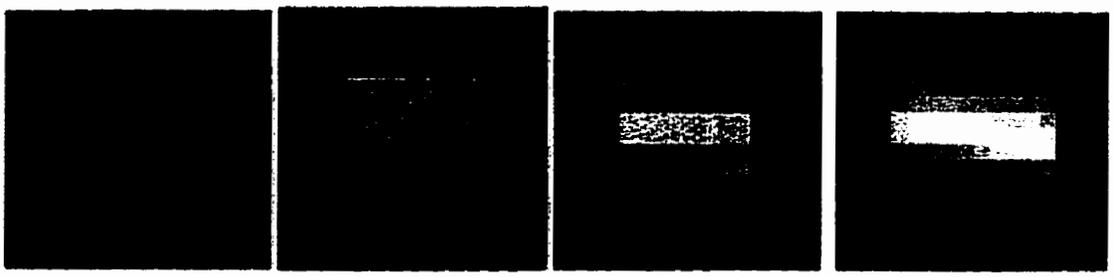
- **Analysis.** The original data or function is decomposed to a linear combination of wavelet basis functions. This gives a representation at the various levels from coarse to fine.

- **Selection.** According to the requirement of resolution and quality metrics, a subset of wavelets is chosen for reconstruction. The selection strategies are the methods to make the choice.

- **Synthesis.** A new approximated data or function is reconstructed from the chosen wavelet subset.

In this chapter we will concentrate on the selection strategies. Figure 4.1 shows the selection of the endpoint-interpolating B-spline wavelets for 2^7*2^7 tooth image. Figure 4.1 (a) shows the Haar wavelets (b) linear B-spline wavelets (c) quadratic B-spline wavelets (d) cubic B-spline wavelets. 2^{level} is the number of coefficients to be selected. The minimum level is 1 and the maximum level is 7 in the Figure 4.1. The pictures in the Figure 4.1 demonstrate that high degree B-spline wavelets require few coefficients with the same image quality.

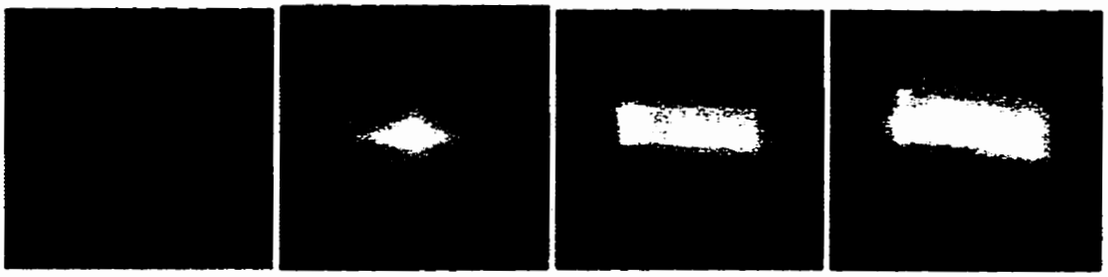
(a) Haar Wavelets (Degree = 0)



Level 1 2 3 4

Level 5 6 7

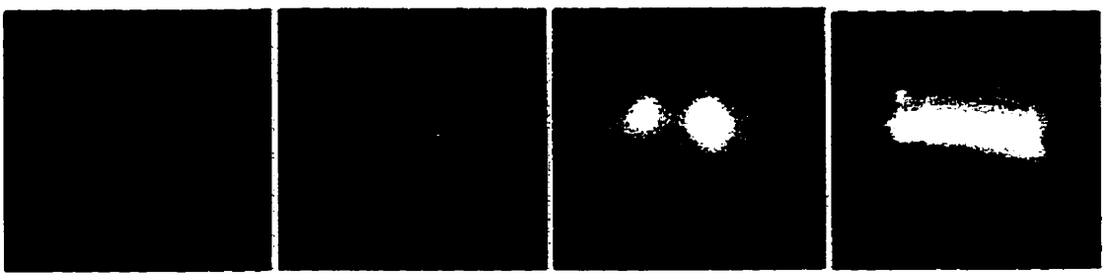
(b) Line B-spline Wavelets (Degree = 1)



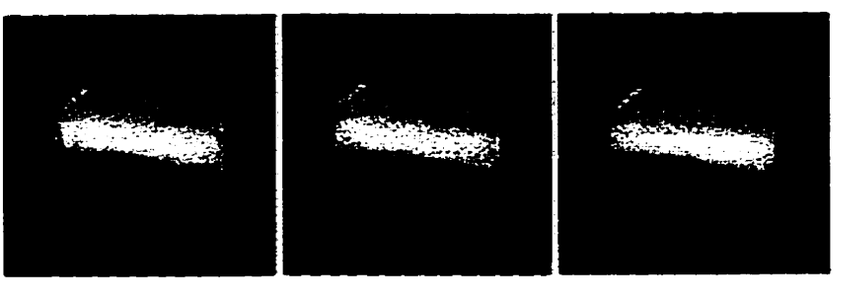
Level 1 2 3 4

Level 5 6 7

(c) Quadratic B-spline Wavelets (Degree = 2)

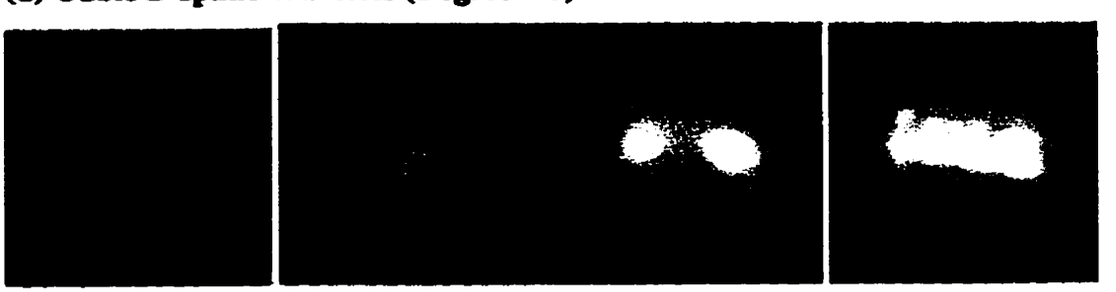


Level 1 2 3 4

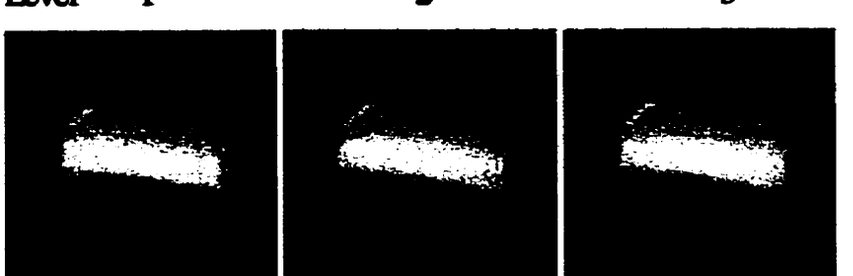


Level 5 6 7

(d) Cubic B-spline Wavelets (Degree = 3)



Level 1 2 3 4



Level 5 6 7

Figure 4.1 Endpoint-interpolating B-spline wavelets for 2^7*2^7 tooth image

4.1 Threshold Testing

Suppose a function f(x) is represented by the linear combination of the coefficient vectors V^0, W^0, W^1, ..., W^{j-1} where j is the level of wavelets, V^0 is the coarse coefficient vector, and W^i for i = 0 to j-1 is the wavelet vector. The simplest selection strategy is to choose wavelet coefficients whose magnitude is greater than a given positive value τ; others coefficients are set to zero. In mathematical terms, suppose $w_i \in W^i$, the new set W'^i is defined as $w_i \in W^i$ such that $|w_i| \geq \tau$ for some given $\tau > 0$.

There are several methods to select τ. One of them is to find the minimum and the maximum magnitude values in the coefficients, and then choose the average to be the τ. In a more general case suppose the minimum and maximum magnitude values are w_{min} and w_{max} respectively, $\tau = \lambda\, w_{max} + (1-\lambda)\, w_{min}$, $0 \leq \lambda \leq 1$. If $\lambda = 0$ then $\tau = w_{min}$ all the coefficients are selected; if $\lambda = 1$ then $\tau = w_{max}$ only the maximum coefficient is selected; when $\lambda = 1/2$, $\tau = (w_{min}+w_{max})/2$ is the average of the w_{min} and w_{max}. The run time for threshold selection is O(n) where n is the number of input data.

4.2 L^2 Progressive Refinement

The compression problem can be explained as follows. Suppose a function is expressed by the scaling functions as

$$f(x) = \sum_{i=0}^{M-1} c_i \phi_i(x)$$

where c_i are the coefficients, $\phi_i(x)$ are the basis functions, and M is the dimension of the vector space that f(x) belongs to. Given an error tolerance $\varepsilon > 0$ we hope to find a function

$$g(x) = \sum_{i=0}^{N-1} d_i \Psi_i(x)$$

such that $N < M$ and $\|f(x) - g(x)\| < \varepsilon$ for some norm.

The L^p norm is defined as

$$\|f(x) - g(x)\|_{L^p} = \left(\int_{-\infty}^{+\infty} (f(x) - g(x))^p \, dx\right)^{1/p}$$

$0 < p < +\infty$. When $p = 1$, it is called L^1 norm; if $p = 2$, it is known as L^2 norm; and for $p = +\infty$, L^p is called L^∞ norm.

Suppose we choose L^2 norm to measure the error and the wavelet basis functions are orthonormal, there is a theorem that says in order to minimize the error for a given N, the best choice is to select the N largest coefficients. See [Stollnitz et al. 1994] for more details.

For the L^2 progressive compression, simply sort the coefficients in order of decreasing magnitude $|c_{\sigma(0)}| \geq |c_{\sigma(1)}| \geq \ldots\ldots \geq |c_{\sigma(M-1)}|$ and then find the smallest N such that $c_{\sigma(N)}^2 + c_{\sigma(N+1)}^2 + \ldots\ldots + c_{\sigma(M-1)}^2 \leq \varepsilon^2$ for a given ε. σ is a permutation of (0 ... M-1). Sorting the coefficients requires minimum O(nlog(n)) operations. Several algorithms are known to speed up this selection strategy [Stollnitz et al. 1994].

In the table 4.1 we define compression ratio as the quotient of the total number of wavelet coefficients and the number of selected wavelet coefficients. It measures the loss of the information. Generally, high compression ratio leads to coarse representation, and low compression ratio consists of more information in details.

L^2 Selection Wavelet Table

	Compression Ratio				Number of Selected Coefficients			
degree / error	0	1	2	3	0	1	2	3
0.5%	7	25	53	43	2128	660	316	399
1%	7	59	131	112	1130	281	129	152
2%	32	195	318	336	505	85	53	51
3%	60	405	603	536	273	41	28	32
4%	99	616	938	746	164	27	18	23
5%	156	875	1300	953	105	19	13	18
10%	655	2773	2414	1430	25	6	7	12

Table 4.1 Compression ratio of the tooth image with L^2 selection for B-spline wavelets of degree 0, 1, 2, and 3.

The Haar wavelet has the smallest compression ratio, but the quadratic B-spline has biggest compression ratio from the Table 4.1. A quadratic B-spline surface may fit the tooth surface better than a cubic B-spline surface.

Generally when a relative error is fixed, a high degree B-spline wavelet approximation needs fewer wavelet coefficients than a low degree B-spline wavelet approximation. That is to say a cubic wavelet with a small number of coefficients may be better than a Haar wavelet with many coefficients. The series of tooth images in Figure 4.1. shows this effect.

4.3 Maximum Error

A problem with the L^2 norm is that the error can be arbitrarily large in a very small area. The L^∞ norm ensures that no part of the error in the reconstruction is greater than a given ε. Refer to [Stollnitz et al 1994] for the algorithm. [DeVore et al. 1992] suggest that L^1 norm is best for the image compression. Please refer to their paper for details.

4.4 Location

In some applications we know the information of some subareas of a surface is more important than others. Therefore we may focus on some areas and ignore others. For the wavelet transformation the correspondence between the wavelet coefficients and the input data should be studied. In Haar wavelets there is a simple algorithm that gives the correspondence. I am going to present it in the following sections.

4.4.1. One-dimensional Haar Wavelets

Suppose an original one-dimensional input vector looks like this

$$I^n = [a_0, a_1, \ldots\ldots, a_{m-1}] \qquad m=2^n,$$

the corresponding Haar wavelets has the same number of coefficients.

$$W^n = [c_0, c_1, \ldots\ldots, c_{m-1}] \qquad m=2^n.$$

From the definition of Haar wavelet in Chapter 3.1, there is a binary tree relating the input vector and the wavelet vector. For example, with n = 3 the binary tree looks like

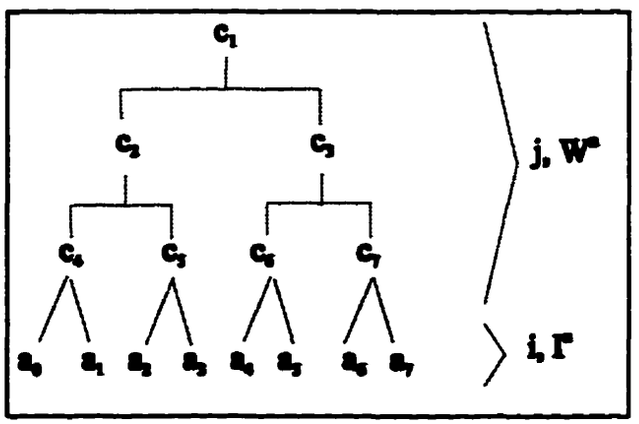


Figure 4.2 Binary tree of Haar wavelet coefficients

The coefficient c_0 is related to all the wavelet coefficients and vice versa, so there is no need to show it on the diagram.

- The main problem is given an index i in I^n, find all the dependent indices j in the W^n. All the dependent nodes of a_i are the nodes in W^n on a path from a_i to root c_1. In general, the left node and right node of c_i in the W^n have the indices 2*i and 2*i+1 respectively. Therefore we can obtain the dependent indices j by the following algorithm:

$$x_0 = \lfloor i/2 \rfloor + 2^{n-1},\ x_1 = \lfloor x_0/2 \rfloor,\ \ldots\ldots,\ x_j = \lfloor x_{j-1}/2 \rfloor,\ \ldots\ldots$$

  The set $\{x_0, x_1, \ldots, 1\}$ includes all the indices j that is corresponded to i in the I^n. For example, let $n = 3$ and $i = 5$, we have $x_0 = 6$, $x_1 = 3$, and $x_2 = 1$, so the dependent index set is [6, 3, 1]. It is the path from c_1 to a_5.

- Another problem is given an index j of W^n, find all the dependent indices i in the I^n. The j of W^n along the leftmost subtree to the leaf in I^n is the starting index, and along the rightmost subtree to the leaf in I^n is the ending index. The other indices

in Γ^n are the indices from starting index to ending index. Let minimum and maximum indices in Γ^n be $index_{min}$ and $index_{max}$, and let $diff$ be the difference of level from node i to the Γ^n level, we have $diff = n - \lfloor \log_2(j) \rfloor$. From the correspondence between the W^n and Γ^n, we have $index_{min} = 2^{diff} * j - 2^n$ and $index_{max} = 2^{diff} * (j+1) - 1 - 2^n$. So given an index j in W^n, the dependent indices are positive integers in the $[index_{min}, index_{max}]$. For example, let $n = 3$ and $j = 3$, then $index_{min} = 4$ and $index_{max} = 7$.

4.4.2. Two-dimensional Haar Wavelets

The correspondence between the two-dimensional input image and wavelets depends on the wavelet transform. For the tensor product wavelet transform which computes wavelet transform on all the rows and then on all the columns, the dependent indices are shown on Figure 4.3. In the figure, suppose original region is the square at the left corner, the dependent regions are represented by shadow. Figure 4.4 and 4.5 illustrate the non-standard and non-standard II wavelets.

For an arbitrary topological region on the rectangular domain, the region can be considered as the set of points. With the mapping of point to points (1 to n) it is easy to obtain the regions of correspondence. Generally a continuous original region may map to discontinuous regions of wavelets. In our special examples, the total number of points in the region is invariant. This property greatly reduces the cost of reconstructing the original data from wavelets. It also saves the space of wavelets for a particular subarea.

The 2D mapping algorithm of tensor product wavelets is directly derived from 1D mapping algorithm. But the 2D mapping algorithms of non-standard wavelets are complicated.

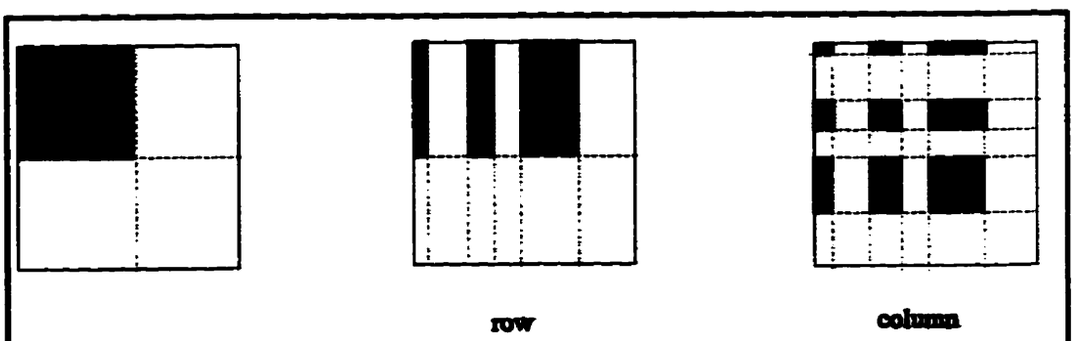


Figure 4.3 Location of standard tensor product wavelets

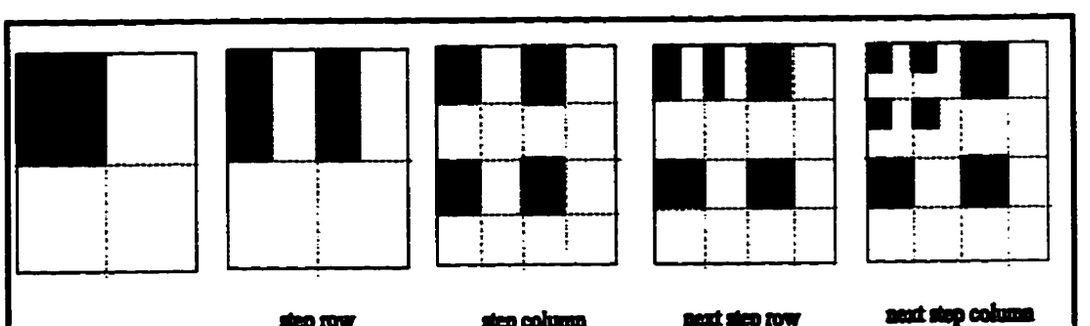


Figure 4.4 Location of non-standard wavelets

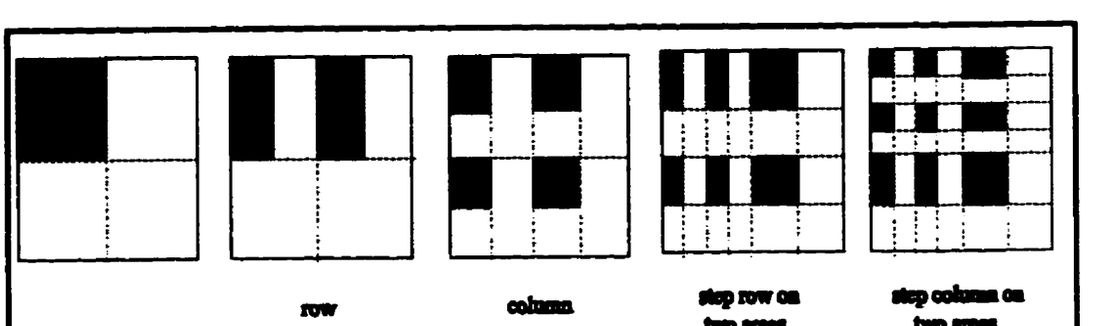


Figure 4.5 Location of non-standard-II wavelets

4.4.3 One-dimensional Endpoint-Interpolating B-spline Wavelets

In this section we will discuss the algorithm to find the correspondence between the input data and the wavelets with endpoint-interpolating B-spline wavelets of general degree. The basic idea is quite simple: based upon the synthesis matrices P and Q or the analysis matrices A and B pick up the related items and then repeatedly apply them to all the filter bank. The following example of cubic B-spline wavelet shows the algorithm.

Given an item and its position in the one-dimensional array we try to find all the wavelet coefficients that are related to the source item. Suppose the (P Q) is the cubic endpoint-interpolating B-spline matrix and the level j starts from 4. The shadow boxes represent non-zero items in the matrix (P Q), the dark shadow boxes are the demonstration of the example, and the blank boxes represent zero items. The (P Q) matrix represents a relation matrix where non-zero items are related, as illustrated in Table 4.2.

The c_{10}, c_{11}, c_{12}, and c_{13} are final coarse coefficients, and the d_{ji} are the wavelets on level j. For example, suppose we want to find all the wavelet coefficients related to the item c_{57} , in step 1) we find the c_{57} from the first column, the corresponding non-zero items are $\{c_{43}, c_{44}, c_{45}\}$ and the wavelets $\{d_{40}, d_{41}, d_{42}, d_{43}, d_{44}, d_{45}\}$. We continue to find the wavelets related to the items $\{c_{43}, c_{44}, c_{45}\}$ in step 2) and get the items $\{c_{31}, c_{32}, c_{33}, c_{34}\}$ and the wavelets $\{d_{30}, d_{31}, d_{32}, d_{33}\}$. Repeat this procedure in step 3) and 4) and we find all the other wavelets $\{d_{20}, d_{21}, d_{10}\}$. And $\{c_{10}, c_{11}, c_{12}, c_{13}\}$ are related to any item of input data. Note the only unrelated wavelet items for c_{57} are $\{d_{46}, d_{47}\}$ in this example.

1) Cubic B-spline (P Q) matrix with j=4

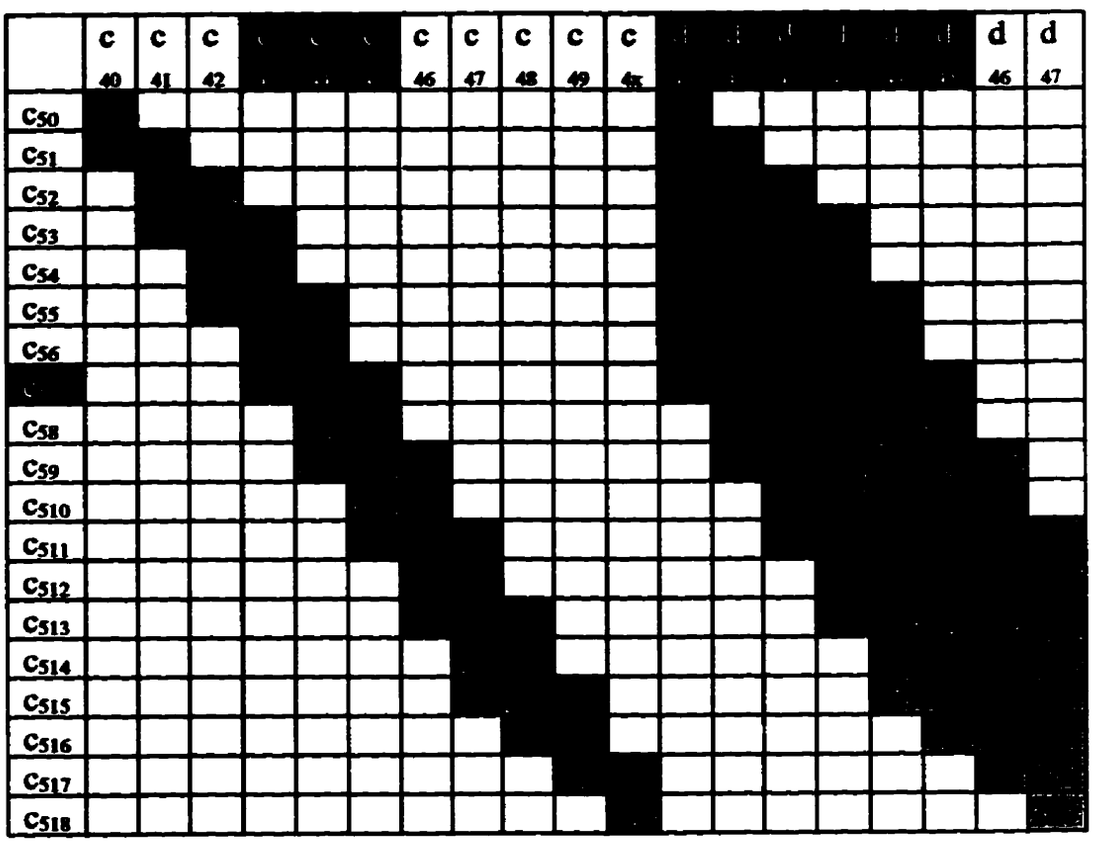


Table 4.2 Cubic B-spline (P Q) matrix with j = 4

2) Cubic B-spline (P Q) matrix with j=3

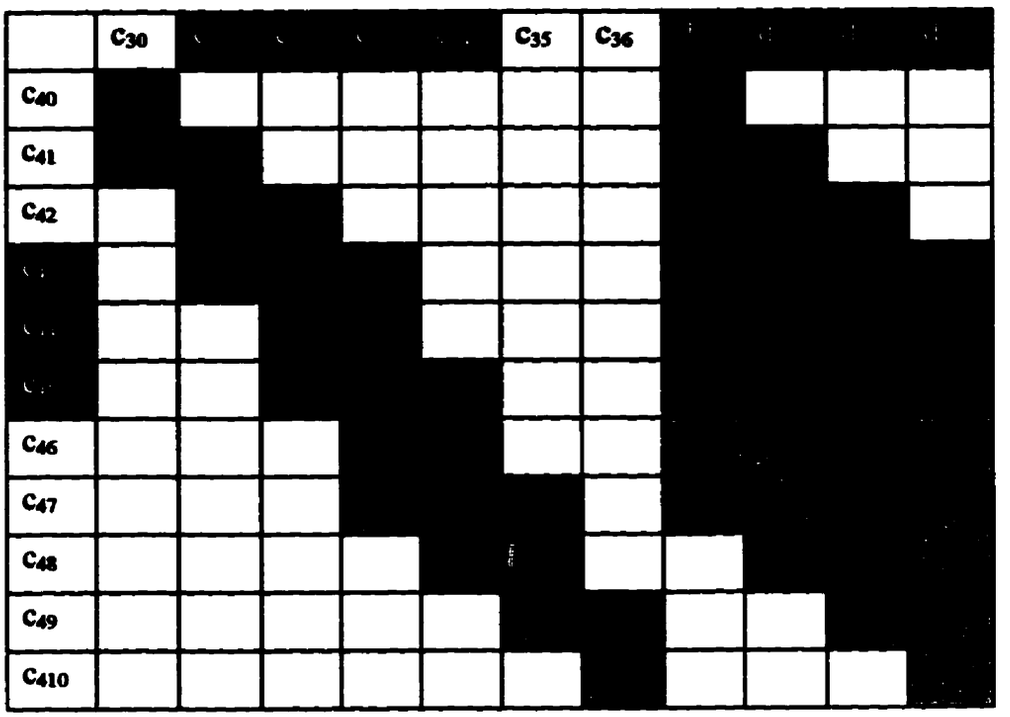


Table 4.3 Cubic B-spline (P Q) matrix with j = 3

3) Cubic B-spline (P Q) matrix with j=2

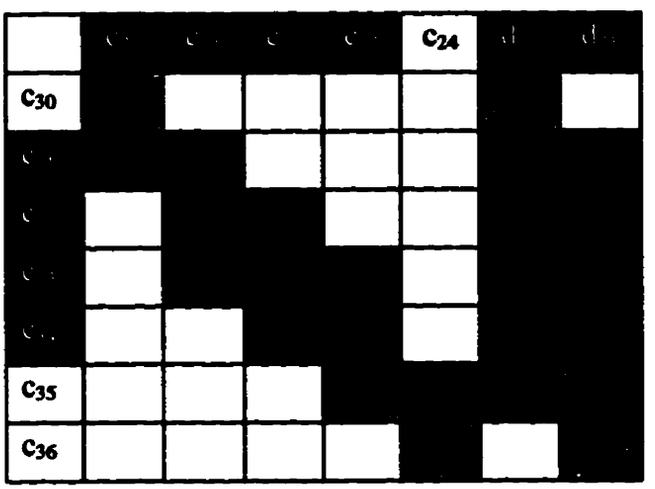


Table 4.4 Cubic B-spline (P Q) matrix with j = 2

4) Cubic B-spline (P Q) matrix with j=1

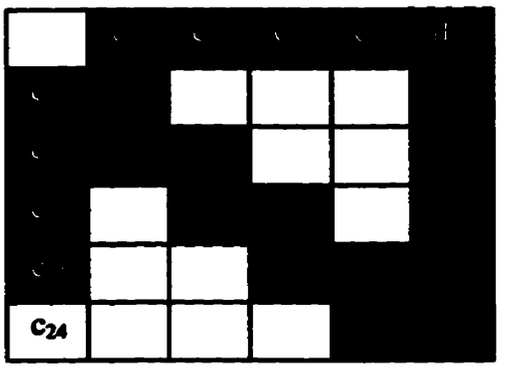


Table 4.5 Cubic B-spline (P Q) matrix with j = 1.

To find dependent input data items from a given wavelet coefficient we need analysis matrix AB which is the inverse matrix of (P Q).

Chapter 4 Selection Strategies

Notes:

I have tested the curvature selection strategy based on the mapping algorithm with Haar wavelets. It works as follows

1. Compute the Gaussian curvature of all the sampling points on the surface.
2. Select a threshold value for curvature, sort data points into low and high curvature by checking if the magnitude of curvature at a point is smaller or bigger than the threshold.
3. Do wavelet transform on input data and get the wavelet coefficients.
4. Find the corresponding wavelet subset for each point with the mapping algorithm. Select wavelet coefficients according to the curvature of a point. For example, we may choose more coefficients for a high curvature point.
5. Reconstruct the surface using the selected wavelets that correspond to selected curvature set.

My experiments showed that the curvature selection with this method has a small compression ratio compared to L^2 selection. The main reason is that the mapping function is 1-n. One way to improve the compression ratio is not to select all the wavelets that correspond to the point but it is difficult to decide which coefficients should be eliminated. For example, a high curvature point maps to one set of wavelets, and a low curvature point maps to another set of wavelets. It is difficult to decide if the intersection set of two wavelets should be selected. In the point mapping algorithm, we noticed that the neighboring points are not gathered into a region. Because of the local spatial property of curvature it is better to separate the points into groups of regions by curvature.

My conclusion is that selection based on curvature without considering its spatial position will not produce good result. That is we can not simply use a sort algorithm

on the magnitude of curvature and just pick up the largest ones. In the next chapters I will discuss the use of curvature for wavelets with subdivision.

4.5 Closeness Measurement

It is possible to do point-to-point comparison between the input data and the approximation output data. A standard measurement is the L^p norm. L^∞ norm specifies the upper bound of the distance between the corresponding points. L^2 norm measures the sum of the squares of these distances. Another widely used measurement is the PSNR (Peak Signal to Noise Ratio),

$$PSNR = 20\log_{10}\frac{1}{\sqrt{\sum_{i=0}^{N-1}(a_i - b_i)^2 / N}}$$

where a_i is the value of original point i; b_i is the reconstructed value of point i; N is the total number of points in the original data. Here we suppose the peak value is 1.

Chapter 5 Curvature Estimation

5.1 Curvature of a Curve Represented by Discrete Points

The material of differential geometry about curves and surfaces is presented in Appendix A. The definition of curvature is also given there. In the following paragraph, I will show how to estimate curvature of a curve represented by discrete points based on the circle that passes through the three points.

5.1.1 The Circle Through Three Points.

First we find the radius of the circle through the three points. This is a neat algorithm with surprisingly small relative errors.

Given three points P_0, P_1, and P_2 the curvature at P_1 can be estimated by finding the circle through the three points. The value of curvature is approximately equal to the reciprocal of the radius of the circle. From geometry this circle is unique. Refer to Figure 5.1.

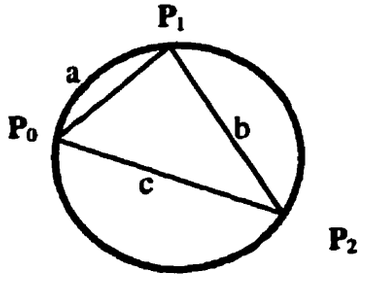


Figure 5.1 Three Point Circle

Suppose the lengths of three edges of the triangle are a, b, c, let s = (a+b+c)/2, from Heron's formula, the area of the triangle $P_0P_1P_2$ is $S = \sqrt{s(s-a)(s-b)(s-c)}$. The

radius of the interpolating circle of three points is $R = \frac{abc}{4S}$ [CRC 88], and the approximate curvature at P_1 is $k = \frac{4S}{abc}$. Another way to compute the area which also gives the correct sign of the curvature is $S = \det\begin{pmatrix}(P_1 - P_0)_x & (P_2 - P_1)_x \\ (P_1 - P_0)_y & (P_2 - P_1)_y\end{pmatrix}$ where the x, y subscripts denote x and y components.

5.1.2 Test Result with Cubic Bezier Curve

I selected a cubic Bezier curve to assess the accuracy of the method of three point circle.

The cubic Bezier curve is

$$P(t) = (1-t)^3 B_0 + 3t(1-t)^2 B_1 + 3t^2(1-t)B_2 + t^3 B_3$$

where B_i are its *control vertices*. Its 1st, 2nd derivatives and curvature are easy to evaluate from above formula. Following is the numerical results of the three point circle algorithm.

For example, consider a Bezier curve with control vertices $\{B_0, B_1, B_2, B_3\} = \{(2.0, 3.0), (5.0, 8.0), (6.0, 3.5), (7.5, 1.5)\}$; the initial step $h = 0.04$ that is continuously reduced to half in the next loop; the parameter ranges from $t_1 - h$ to $t_1 + h$ where $t_1 = 0.185$; the estimating 3 points are P(t-h), P(t), P(t+h) and the curvature is estimated at P(t); the $Error_i$ is *(estimating_value - exact_value)/exact_value*; the Error Ratio is $Error_{i+1}/Error_i$.

Step	Exact	3Point Circle	Error	Error Ratio
	0.344172		-0.00307785	
	0.344172		-0.00077834	
	0.344172		-0.00019457	
	0.344172		-0.00004771	
	0.344172		-0.00001190	
	0.344172		-0.00000298	
	0.344172		-0.00000075	
	0.344172		-0.00000019	
	0.344172		-0.00000005	
	0.344172		-0.00000002	

Table 5.1 Curvature Estimation with three Points Circle

Conclusion:

The step $h_{i+1} = h_i/2$ and the $Error_{i+1} = Error_i/4.0$, which suggests the relative error is $O(h^2)$.

5.2 Estimation of Gaussian Curvature

In the case of a surface, Gaussian curvature is an intrinsic property of a surface at a point like the curvature of a curve at a point. Gaussian curvature is defined as the product of the principal curvatures, and the directions giving the two principal curvatures are mutually orthogonal. Refer to Appendix A for more details. [Stokely & Wu 92] reviewed several methods for estimating the Gaussian curvature. [Koenderink 92] presented a method to describe a surface shape with curvature. [Lee et al 93] gave a scheme to estimate the curvature from sampling noisy data. [Todd 86] described numerical estimation of the curvature of surfaces.

The method (proposed by Dr. Meek) to estimate the Gaussian curvature by normals is completely different from the methods mentioned by the above authors. We estimate the Gaussian curvature by the approximation of normals and the area of spherical image. I tested the Surface Triangulation (ST) method mentioned in the paper [Stokely & Wu 92] and found it does not converge. The method to estimate the Gaussian curvature by normals is more stable and fast to converge.

5.2.1 Estimation of Gaussian Curvature with Normals

A *valence* of a vertex on a polyhedron is defined as the number of faces around the vertex. If we triangulate a rectangular grid, each vertex will have valence 4 or 8, see Figure 5.2. For convenience, the boundary and corner points are copied and extended to one more row and column.

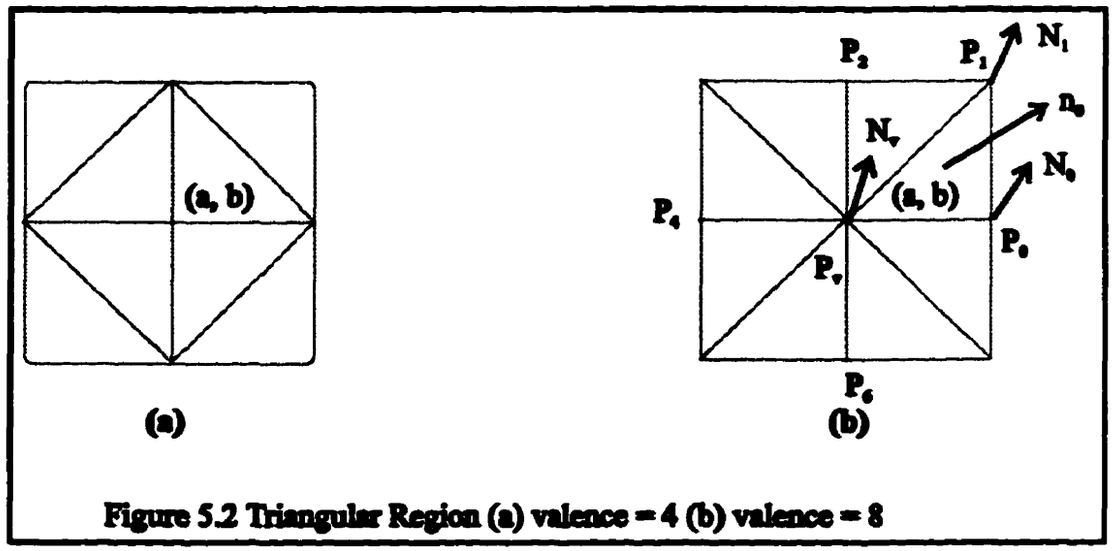


Figure 5.2 Triangular Region (a) valence = 4 (b) valence = 8

In the next section we will describe the estimation of Gaussian curvature based on the normals on a grid.

Let m be the valence at a vertex v. N_v is the unit normal at the vertex v. The neighbor points of v are P_0, P_1, ..., P_{m-1} in counter-clockwise order from P_0. $S_{P_0P_1P_v}$ is the signed area of planar triangle $P_0P_1P_v$. n_i is the unit normal of the planar face $P_iP_{i+1}P_v$, and is equal to the normalized cross product of vectors P_vP_i and P_vP_{i+1}. It is $\frac{P_vP_i \times P_vP_{i+1}}{\|P_vP_i \times P_vP_{i+1}\|}$.

Refer to Figure 5.2. The algorithm to estimate Gaussian curvature is

- **Get the normal estimation at vertex v.**

$$M_v = \frac{S_{P_0P_1P_v} * n_0 + \ldots\ldots + S_{P_{m-1}P_0P_v} * n_{m-1}}{S_{P_0P_1P_v} + \ldots\ldots + S_{P_{m-1}P_0P_v}},$$

then $$N_v = \frac{M_v}{|M_v|}.$$

- **Find the normals at all grid points.**

In the same way, we can get N_0, N_1, ... , N_{m-1} the approximate unit normals at each vertex. In order to compute the normals of boundary vertices, we can copy the boundary vertices to extend the boundary.

- **Compute curvature from the normals**

Imagine that N_0, N_1, ... , N_{m-1} form the spherical image on the unit sphere. The area of spherical image can be estimated by the sum of $S_{N_0N_1N_v} + \ldots\ldots + S_{N_{m-1}N_0N_v}$. Where $S_{N_0N_1N_v}$ is the area of the triangle on the unit sphere that has vertices N_0, N_v, N_1 .

The Gaussian curvature can be estimated by the formula

$$K = \frac{S_{N_0N_1N_v} + \ldots\ldots + S_{N_{m-1}N_0N_v}}{S_{P_0P_1P_v} + \ldots\ldots + S_{P_{m-1}P_0P_v}}$$

5.2.2 Test Scheme for Curvature on Surface

In the following paragraph I will describe the estimation of the curvature of a sphere and of the bell-shaped surface.

The basic sampling data is a 5x5 patch when we want to compute the curvature at the centre point (2,2). The sampling distance is represented by h. The valence is set to 8. Figure 5.3 shows the sampling patch.

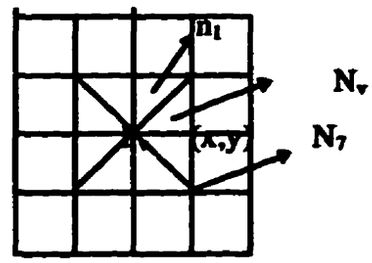


Figure 5.3. Sampling Patch

- The semi-sphere is defined as

$$S(x, y) = (x, y, \sqrt{R^2 - x^2 - y^2}), \quad R > 0.$$

Its Gaussian curvature is $1/R^2$.

- The bell-shaped surface is defined as

$$B(x, y) = (x, y, \exp(-(a*(x-x_0)^2 + b*(y-y_0)^2))), \qquad a, b > 0 .$$

Its Gaussian curvature for $a = b$ and $x_0 = y_0 = 0.0$ is

$$\boxed{K_B(x,y) = \frac{(2a)^2(1-2a(x^2+y^2))\exp(-2a(x^2+y^2))}{(1+(2a)^2(x^2+y^2)\exp(-2a(x^2+y^2)))^2}} .$$

Suppose $r^2 = x^2 + y^2$ ($r \geq 0$), we can replace $K_B(x, y)$ with $K_B(r)$.

$$K_B(r) = \frac{4a^2(1-2ar^2)\exp(-2ar^2)}{(1+4a^2r^2\exp(-2ar^2))^2}$$

$$K_B'(r) = -32a^3 r\exp(-2ar^2)\frac{(1-ar^2)+2a((1-ar^2)^2+a^2r^4)\exp(-2ar^2)}{(1+4a^2r^2\exp(-2ar^2))^3}$$

Conclusions:

- when r = 0, $K_B(0) = 4a^2$ is the maximum of the function $K_B(r)$.
- $K_B(r)$ has a minimum value when $r \neq 0$ and $K_B'(r)=0$.
- when $r \to \infty$ the $K_B(r) \to 0$.
- the sign of $K_B(r)$ is decided by the value a.

$$K_B(r)=\begin{cases} >0 & if\ r < 1/\sqrt{2a} \\ 0 & if\ r = 1/\sqrt{2a} \\ <0 & if\ r > 1/\sqrt{2a} \end{cases}$$

5.2.3 Test Result for Example Surfaces

Definition

	sampling distance in x or y direction
	approximating Gaussian curvature
	exact Gaussian curvature
	relative error = (aGC-eGC)/eGC
	$error_{i+1}/error_i$

- **Sphere**

Point Location (x, y) = (0.0, 0.0)

Radius R = 1.0

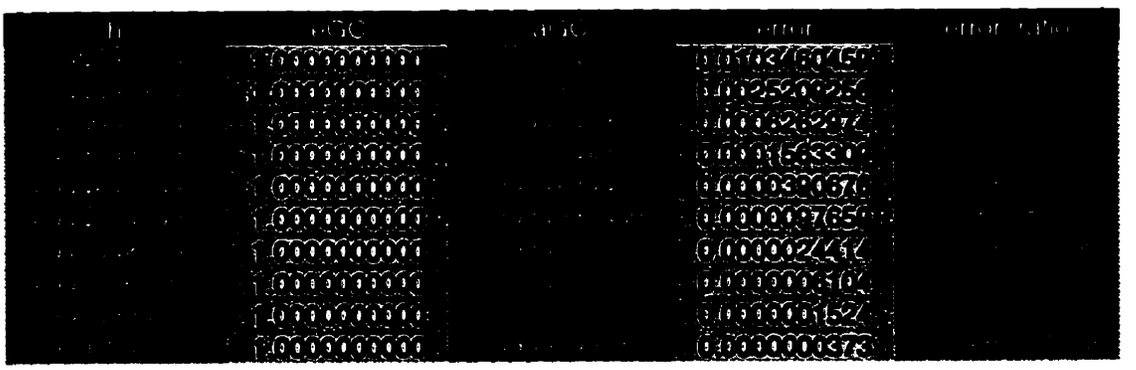


Table 5.2 Gaussian Curvature on Sphere

- **Bell-Shaped Surface**

Point Location $(x, y) = (0.1, 0.1)$

Parameter $a = 0.5$

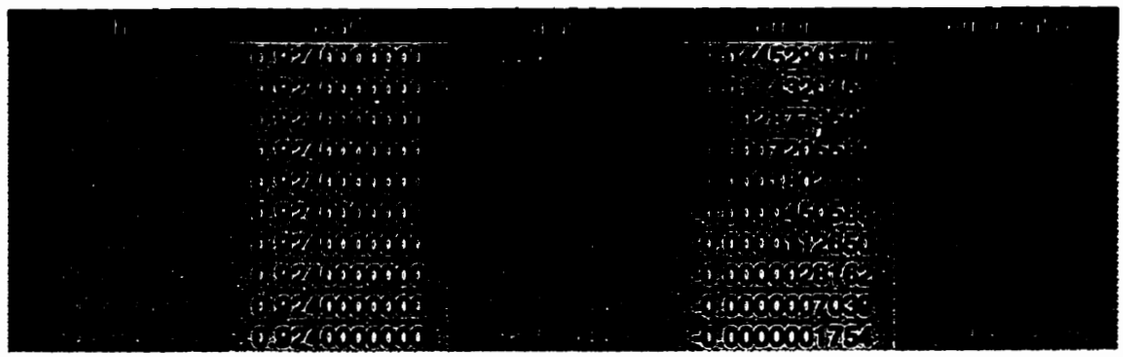


Table 5.3 Gaussian Curvature for Bell-shaped Surface

Conclusion:

The error_ratio is near the 1/4 when h is reduced by half. Therefore the error is $O(h^2)$.

We see the estimation converges well.

5.3 Curvature Measurement

An approximation scheme should keep important features of the input data. Generally to represent curves and surfaces the areas of sharp variation need more information. A high curvature indicates a sharp variation near a point on a curve or a surface. For approximation of a surface, areas of high curvature require more patches, while larger flat regions with small curvature need fewer patches.

The L^2 measurement of curvature K(A, B) between the reconstruction surface and original surface is defined as

$$K(A,B) = \sqrt{\frac{\sum_i (a_i - b_i)^2}{N}} \quad .$$

A is the original surface, a_i is the curvature at the point i of A. B is the reconstructed surface, b_i is the curvature at the point i of B. N is the total number of sampling points of A.

Chapter 6 Subdivision Based on Curvature

6.1 General Study

Subdivision is a well-known technique in Computer Graphics [Cohen et al 80] and [Foley & Van Dam 92]. In the paper [Hamann 94] Hamann described a scheme for compact triangulation of a surface. In this section suppose we are dealing with the region data surface which is defined in a rectangular region called a *window*, the main idea of curvature selection by subdivision is as follows:

First we compute the curvature at each point in the *main window*. By curvature criteria we decide whether to subdivide the window into a series of sub-windows with quad-subdivision. In every sub-window, we apply wavelet transforms and select the coefficients to get a compact representation. Finally we can reconstruct the surface with the selected wavelets.

A subdivision algorithm for region data defined on the rectangle field works as follows:

Suppose the rectangle area are divided into four subareas represented by A_0, A_1, A_2, A_3. See Figure 6.1. Let *threshval* be the threshold value of curvature.

Figure 6.1 Subdivision

```
CSubdivision (rect, threshval)
{
        if (rect is the minimize size) {
                record the rect.
                return;
        }

        if (all the curvatures in the rect are smaller than the threshval) {
                record the rect.
                return;
        }
        CSubdivision( rect.A0, threshval );
        CSubdivision( rect.A1, threshval );
        CSubdivision( rect.A2, threshval );
        CSubdivision( rect.A3, threshval );
}
```

This subdivision algorithm divides the window into sub-windows represented by a quad-tree. Each leaf of the tree represents a non-overlay window. The depth of the subdivision quad-tree is an index that indicates the variation of curvature.

6.2 Local Haar Wavelets (LHW)

In the Chapter 3 I have described the standard and non-standard wavelets. Here I will introduce Local Haar Wavelets (LHW).

A subdivision window is represented by W(i,j), where i is the level and the root has level 0; j is the window's serial number, it is a number from 0 to 3 for the quad-subdivision. For example, W(0,0) is the root, W(1,0), W(1,1), W(1,2), W(1,3) are the first level windows. The sequence of Window W(i,j) can be represented in the dictionary order of (i, j). The position and size of a window can be concluded from the window sequence. We call top-left element in a window W(i,j) the *amplitude* and represent it as Amp(W(i,j)). If there is only one element in the window the Amp(W(i,j)) is equal to the value of that element. In the following discussion I suppose the window is square and its size is the number of the row or column elements.

For quad-subdivision,

$$\boxed{W(i-1,l) = \bigcup_{j=0}^{3} W(i,j)} \qquad i > 0.$$

where l is the serial number of parent window in i-1 level.

Let LHW (W(i,j)) represent Local Haar Wavelets on the window W(i,j), then the Local Haar Wavelets is defined by a recursive relationship,

$$LHW(W(i-1,l)) = \begin{cases} Amp(W(i-1,l)) & \text{if the size of window } W(i-1,l) \text{ is 1} \\ HW(Amp(LHW(W(i,0))), \\ \quad Amp(LHW(W(i,1))), \\ \quad Amp(LHW(W(i,2))), \\ \quad Amp(LHW(W(i,3)))) & \text{for amplitude elements of } W(i,k) \\ & 0 \le k \le 3 \\ \bigcup_{k=0}^{3} LHW(W(i,k)) & \text{for non-amplitude elements of } W(i,k) \end{cases}$$

where l is 0, 1, 2, or 3. Suppose the four top-left points of a window are represented by the matrix $\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix}$. Its Haar Wavelets *HW* (non-normalized) are

$$HW\begin{pmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{pmatrix} = \begin{pmatrix} \frac{(a_{00}+a_{01})+(a_{10}+a_{11})}{4} & \frac{(a_{00}+a_{10})-(a_{01}-a_{11})}{4} \\ \frac{(a_{00}+a_{01})-(a_{10}-a_{11})}{4} & \frac{(a_{00}+a_{11})-(a_{01}-a_{10})}{4} \end{pmatrix}$$

That is to say the high level of Local Haar Wavelets (LHW) on a window is constructed by picking up 4 amplitude elements of Local Haar Wavelets (LHW) on the four sub-windows and doing the Haar Wavelet transform (HW) on the four amplitudes.

Like non-standard Haar wavelet transform (see Chapter 3), the Local Haar Wavelets (LHW) of 2x2 is given by first applying a one-dimensional wavelet transform on rows (a_{00}, a_{01}) and (a_{10}, a_{11}). The position of (a_{00}, a_{01}) is replaced by $(\frac{a_{00}+a_{01}}{2}, \frac{a_{00}-a_{01}}{2})$ and (a_{10}, a_{11}) by $(\frac{a_{10}+a_{11}}{2}, \frac{a_{10}-a_{11}}{2})$. Then we apply the Haar wavelet transform on all columns and so on. Note the difference between the non-standard Haar wavelets and the Local Haar wavelets is the location of the wavelet coefficients. From the Chapter 3 we know that the higher level of coefficients of the non-standard Haar wavelets are extended from top-left corner to the bottom-right; the coefficients of the Local Haar Wavelets on the other hand are distributed in the location of *row(mod* 2^k*)* and *column(mod* 2^k*)* from coarse to fine. The following diagram is a demonstration of one-dimensional Local Haar Wavelets (LHW).

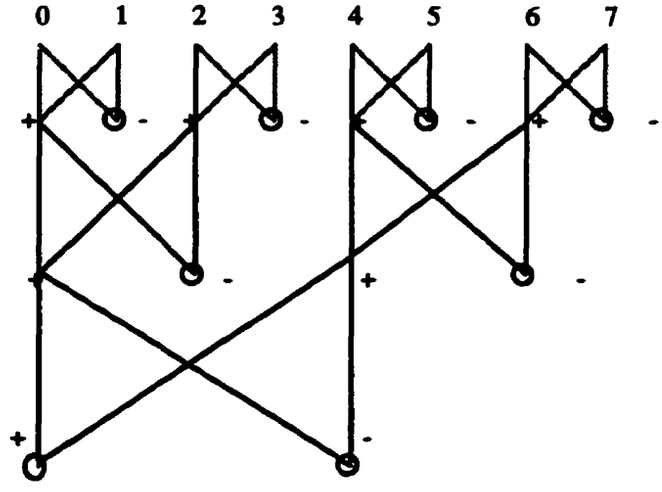


Figure 6.2 Local Haar Wavelets

This architecture is looks like the recursive butterfly in the FFT algorithm. The nodes with only two input edges are the wavelet coefficients.

From the LHW on the entire window, we can get Local Haar Wavelets (LHW) for every subdivision windows. First we compute the LHW for entire window. When we subdivide the windows we use the *Reverse Local Haar Wavelets (RLHW)* to the amplitude elements on the sub-windows. In this way the LHW are produced recursively in the procedure of subdivision. This provides us the flexibility to use wavelets according to the spatial property of a surface.

The subdivision by curvature can be thought of as a kind of coding scheme. The sequence of windows W(i,j) represents the surface by the curvature properties. Each window is represented by several wavelet coefficients.

The LHW is suitable for the subdivision algorithm presented before. We can construct the non-standard Haar wavelets from a sequence of subdivision windows and Local Haar Wavelets of each window. In this meaning the non-standard Haar wavelets are a special case of LHW with only one window. Another advantage of LHW is the spatial localization. The subdivision method first divides a window into sub-windows according to a cost function, such as the flatness of the curvature, and then does the wavelet transforms on the sub-windows.

This procedure links adaptive subdivision by curvature and wavelets. For example we can select a curvature threshold and the relative error to decide the compact representation of the region data. Curvature measures the flatness of a region. A general principle is that in a flat region we can select fewer coefficients to represent it. So we can choose fewer coefficients of wavelets in a "flat" sub-windows and more coefficients of wavelets in other sub-windows.

6.3 Bell-shaped Surface

The subdivision and Local Haar Wavelets (LHW) is decided by a pair of parameters (τ, ε). τ is the value of the *curvature threshold*. It gives global control to the subdivision frame and the size of windows. ε is the *relative error* for a reconstruction window with Local Haar Wavelets. It gives local control to the accuracy of approximation in a subwindow.

Suppose f(x) is an input function and g(x) is the reconstruction function after the wavelet transform applied to f(x). Let $\varepsilon^2 = \|f(x) - g(x)\|^2$, a theorem states that ε^2 is equal to the sum of the squares of unselected wavelet coefficients which have been sorted, refer to [Stollnitz et al 94]. Therefore given ε we can find the selected wavelet coefficients.

When we increase τ and leave ε fixed the number of windows gets small and the size of window becomes large generally. That is the "flat" areas become bigger.

Following data and pictures are done on the Bell-shaped surface defined on Chapter 5.

$$B(x,\ y) = (x,\ y,\ exp(-(a*(x-x_0)^2+b*(y-y_0)^2))$$

with a = b = 8.0 and the centre $(x_0, y_0) = (-1.0, +1.0)$. The sampling data are taken from the rectangular range with the top-left point $(-2.0, 2.0)$ and bottom-right point $(2.0, -2.0)$. The sampling distance is 1/32.0.

Bell-shaped Surface	Subdivision & LHW			Standard Haar Wavelet	
Compression Ratio	τ	Number of Coefficients	PSNR	Number of Coefficients	PSNR
39	3.0715	414	20.472239	412	23.762551
83	200.00	199	20.303242	199	20.152546

Table 6.1 Parameters and results for subdivision and standard wavelets

PSNR is the Peak Signal to Noise Ratio. It is an error measurement to image quality. The definition is given in the Section 4.5.

In the Table 6.1 we see the number of selected coefficients are almost equal for both subdivision with LHW and standard Haar wavelet. The PSNR of Haar wavelet is better in one and worse in the other.

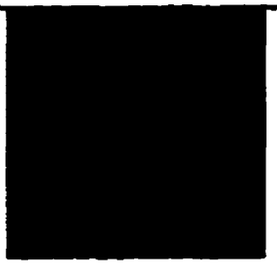
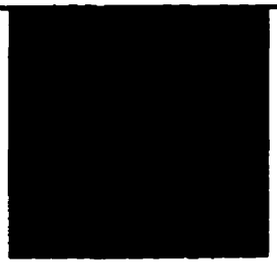
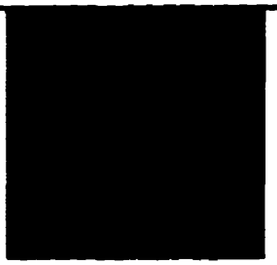
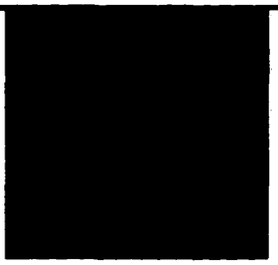
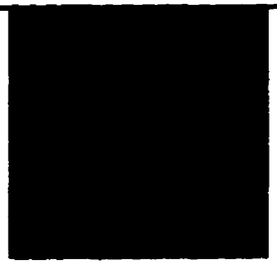
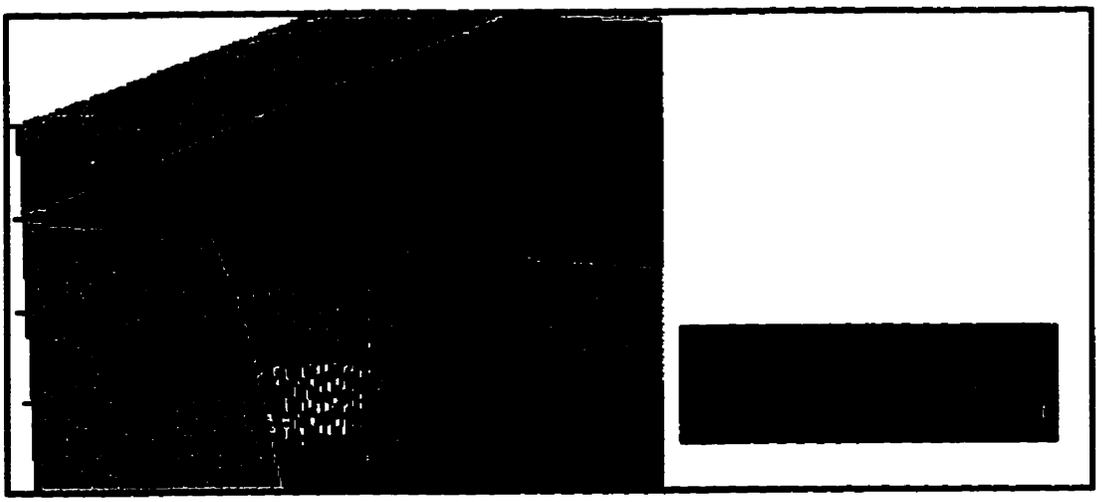
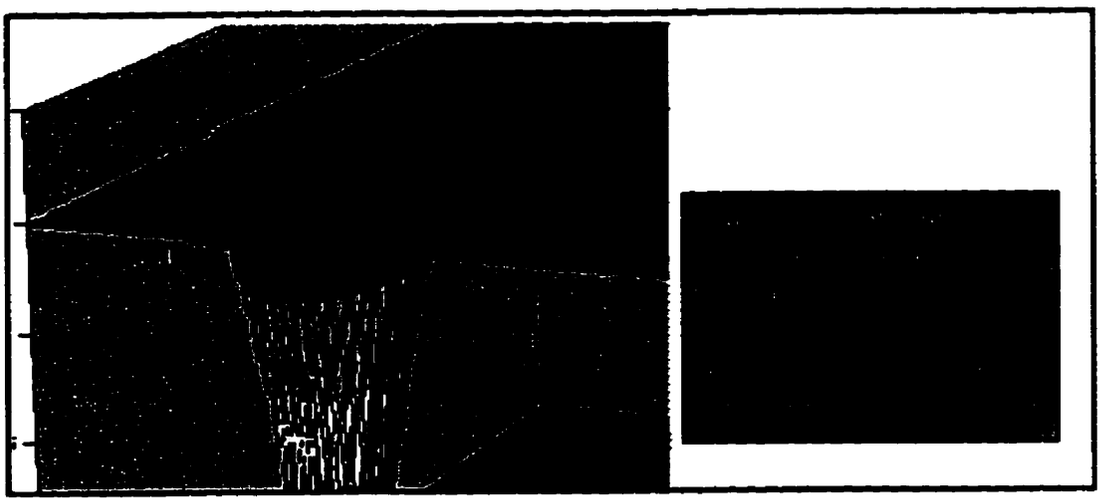
τ	Subdivision Frame	Subdivision & LHW Image	Haar Wavelet Image
3.0715			
200.00			

Figure 6.3 Frame and image pictures

In the Figure 6.3 I show the frame pictures of subdivision by curvature. The small window size areas represent the high curvature areas. The frame pictures correctly show the distribution of the curvature for bell-shaped surface. The image pictures show little distortion compare to the original data for the selection in the Table 6.1.





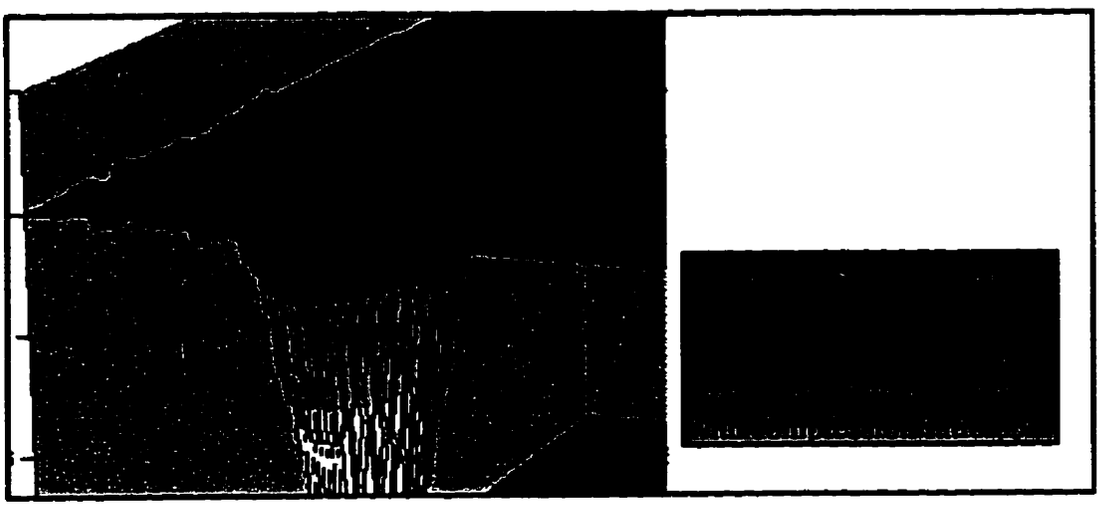
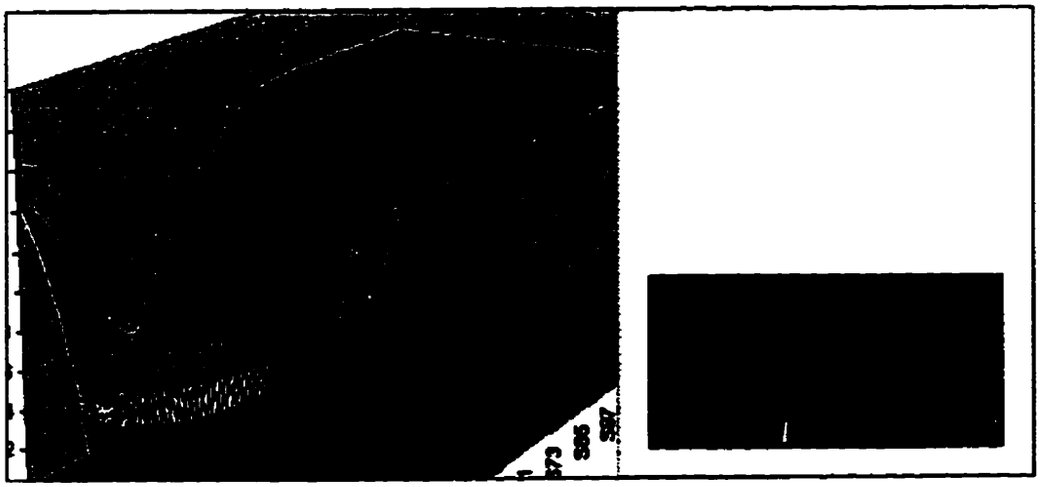
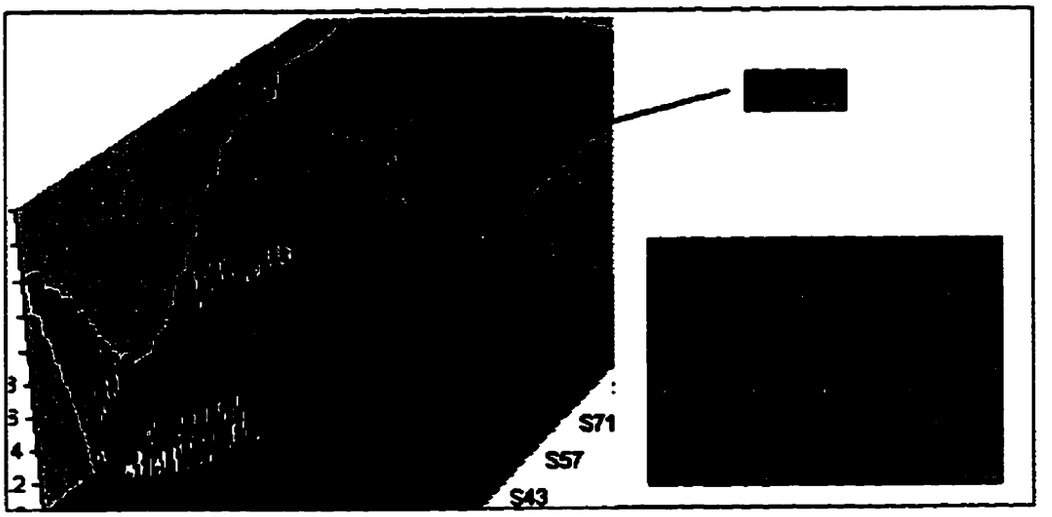
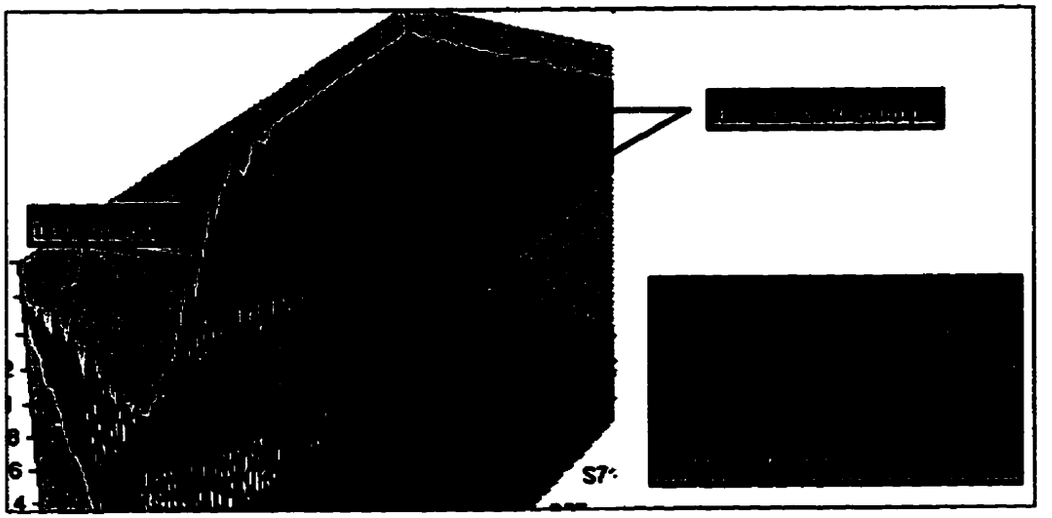


Figure 6.4 is the 3D surface of the original bell-shaped data used on Table 6.1. Figure 6.5 is the 3D surface for the LHW and subdivision based on curvature with compression ratio 83 in the Table 6.1. Figure 6.6 is the 3D surface for the standard Haar wavelet with compression 83 in the Table 6.1.

Comparing these three figures, we find that there are some wrinkles near the flat boundary of the bell-shaped surface in Figure 6.6 for the Haar wavelet. The surface in Figure 6.5 looks closer to the original surface in that area. In the flat area the subdivision based on curvature surface looks more smooth than the standard Haar wavelet surface.




S71
S57
S43



The surface section is obtained by applying the following function

$$f(x)=\begin{cases} minval & x<minval \\ maxval & x>maxval \\ scale*(x-minval) & otherwise \end{cases}$$

onto the input data. $f(x)$ is a linear function so it only changes the view of the surface and does not change the property of the original surface. In the above images I set the *minval* = *0.8*, *maxval* = *1.2*, and the *scale* = *8*.

From Figures 6.7, 6.8, and 6.9 we can see that there are bumps in the flat area of the Standard Haar Wavelet Surface. In the Figure 6.8, the flat area is smooth like the original surface. We see that the LHW with subdivision based on curvature produces better surface.

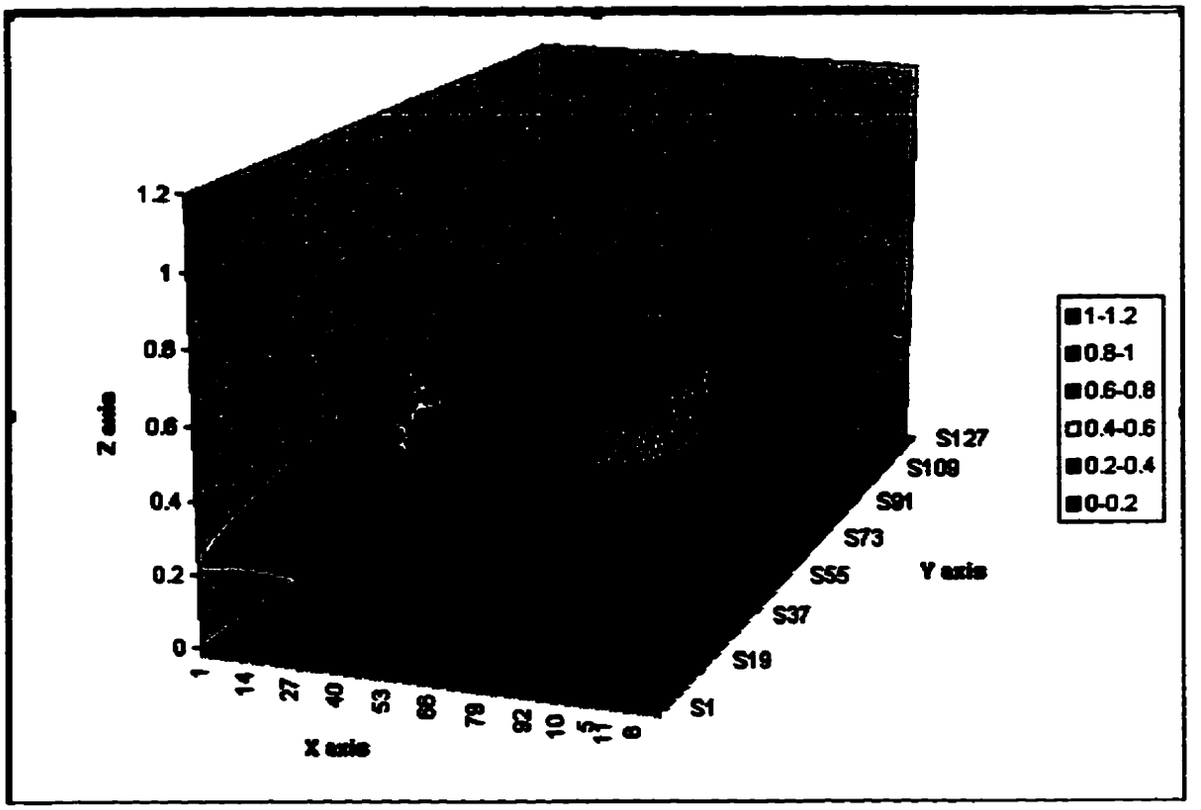


Figure 6.10 Original tooth surface

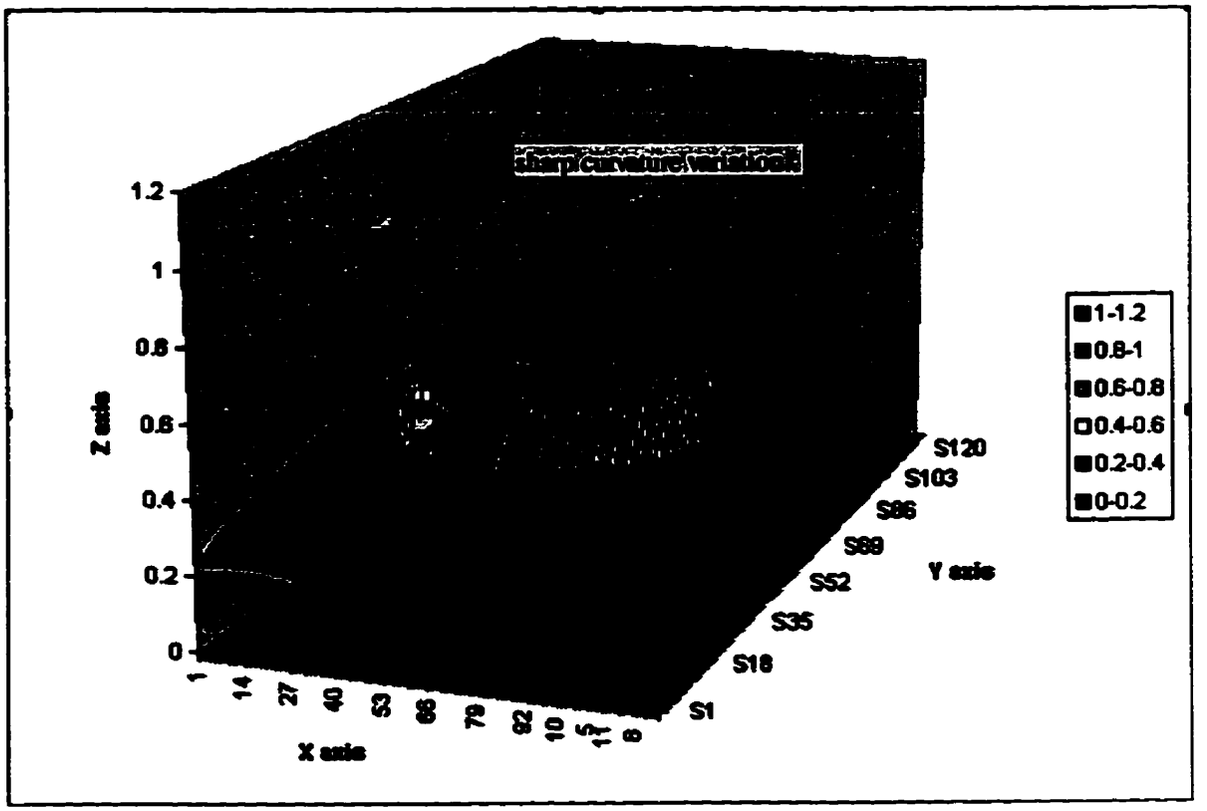


Figure 6.11 LHW and Subdivision Tooth Surface

The parameters and results are listed in the following table.

τ	ε	number of coefficients	compression ratio	PSNR
0.1	0.0078	3079	5.32	19.20

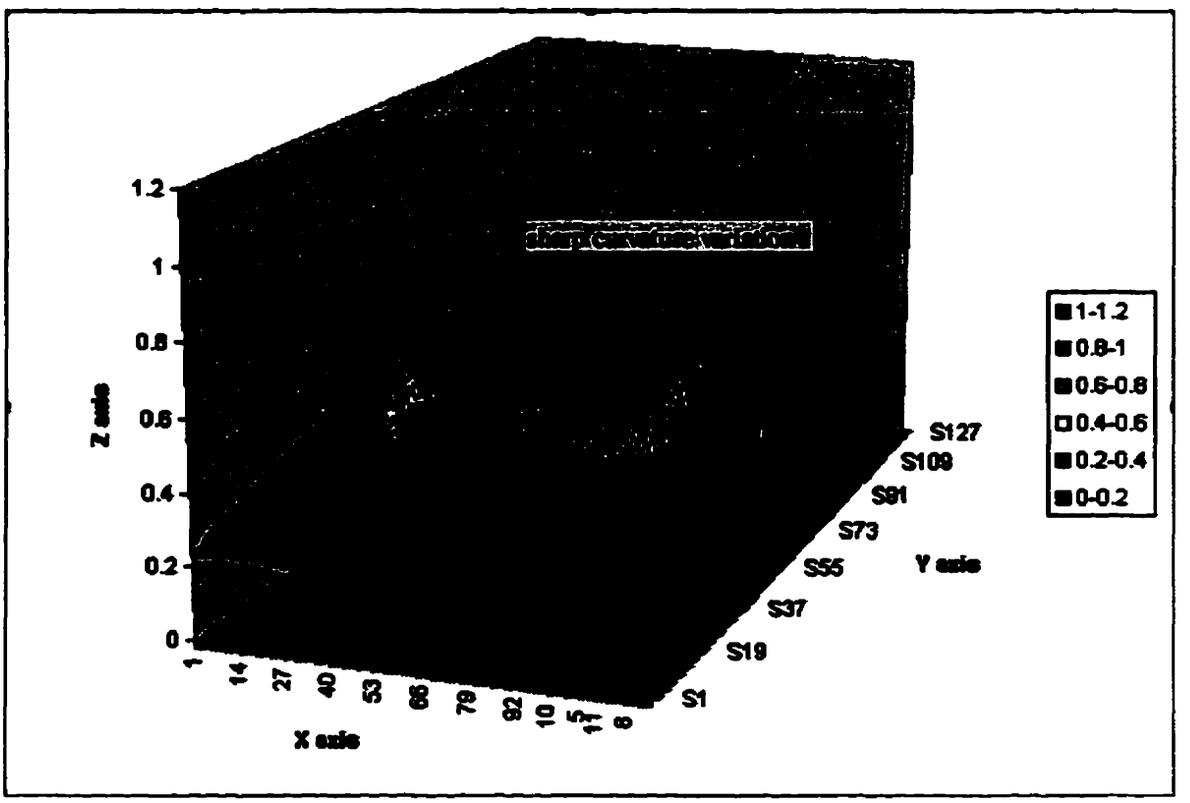


Figure 6.12 Standard Haar Wavelet Tooth Surface

The parameters and result are listed in the following table.

ε	number of coefficients	compression ratio	PSNR
0.003	3091	5.3	29.10

Conclusions for Tooth Image:

From above pictures, Figures 6.10, 6.11, and 6.12, we see both LHW with subdivision based on curvature and standard Haar wavelet surfaces are very close to original surface. Notice both surfaces have the compression ratio 5.3. In a small area with sharp curvature variation pointed to by the arrow, we see the subdivision method is not very satisfactory. One reason may be that poor accuracy in the data leads to large errors in the curvature estimate. Another reason may be the subdivision boundary. The uniform subdivision like quad-subdivision may be not very suitable for the small area with sharp variation. This problem requires more study.

Chapter 7 Conclusions and Future Work

In this dissertation, I have applied a method of wavelets and differential geometry on the surface reconstruction, selection, and measurement. The endpoint-interpolating B-spline wavelets and several selection strategies are discussed. Subdivision based on the curvature and Local Haar Wavelet are also discussed.

7.1 Applications

In previous chapters, I examined the basic theory and algorithms for B-spline wavelets and curvature approximation. It is applied on examples where the surface is given mathematically and sampled as range data. In this section I will summarize some potential applications with this method.

7.1.1 Data Compression

This is the most obvious application for wavelets. With endpoint-interpolating B-splines we can construct the filter bank and complete the wavelet transform in linear run time. The image compression is a special case of surface compression in which the surface is given by height field data. There is already a lot of work related to the image compression with wavelets. The work of [DeVore et al. 92] is well known. Multiresolution curves have been studied by [Finkelstein and Salesin 94]. A data reduction scheme for triangulated surfaces based on curvatures is developed by [Hamann 94].

The wavelet compression based on a curvature metric has the advantage of considering the geometric properties of flatness and smoothness. Curvature represents the local geometric property of the surface. For example, in theory, we can represent the area with the negative curvature by hyperbolic type surface; the area with positive

curvature by elliptic type surface; and the area near zero curvature by parabolic or planar surface.

7.1.2 Progressive Transmission and Rendering

Multiresolution representation of a surface is very suitable for progressive transmission (transmit data from coarse to fine) on the network. In the client site the surface can be rendered in a progressive mode. It is also possible to reduce the time of wavelet surface rendering by using the level-of-detail control such as in radiosity [Gortler et al. 93].

7.1.3 Location Viewing and Image Enhancement

Considering the mapping algorithms presented in this dissertation, we can select interesting regions to reconstruct the surface from wavelets. We can also preview a picture and a map by this method. The mapping algorithm combined with other selection strategies can effectively speed up the reconstruction.

The subdivision wavelets based on Gaussian curvature provide us with a method to enhance an area of an image based on the curvature. For example, some interesting areas are the sharp variation areas which are represented by the large curvature magnitude value. We can select more wavelet coefficients in these interesting areas.

7.1.4 The Quality of Approximation Surface

Given an approximation to a surface, a key problem is to evaluate the quality of the approximation. Global metrics such as PSNR that measure the reconstructed surface quality are not always satisfactory. With the curvature approximation we can compare the curvatures of original and the new one. This provides us a metric to measure the

quality of the reconstructed surface. Since it is not easy to compute the curvatures of a Haar, linear, and quadratic B-spline wavelet surface directly by parametric representation, the approximation method developed in the Chapter 5 to estimate the Gaussian curvatures from discrete points is useful.

7.1.5 Feature Extraction and Match

The lack of translation-invariance is a weakness of orthonormal wavelets. However curvature is translation-invariant. So it is possible to use curvature for finding the translation information. The translation-invariance property is required for applications such as image registration and stereo pair match. The curvature of a surface also provides the important information for feature extraction.

7.2 Conclusions

In this dissertation, I worked on applying the method of wavelets and curvature to solve a problem in computer graphics. I had studied the basic ideas and algorithms. Several programs are designed to test the performance and error of the proposed methods. These programs are implemented under Windows95 and Linux with object-oriented programming language C++ and MFC. To work on the program I also learned a lot of object-oriented and numerical computation methods. For example, I learned how to estimate the accuracy of floating-point numbers in a computation.

7.3 Future Work

One possibility of future work is to extend the method of curvature and wavelets to 3D surfaces with arbitrary topology. New selection strategies may be required to reconstruct and select the surface. The subdivision based on curvature can also be extended to 3D surfaces. The Local Haar Wavelets can be used in the situation where the adaptive subdivision is needed. It is possible to extend the Local Haar Wavelet to high degree wavelets with continuity.

A metric to measure the quality of a surface based on the curvature estimation is needed. Other approaches to measure the quality of an approximation curve and surface scheme based on the differential geometry need to be developed. The curves and surfaces with the curvature constraint condition may be more smooth.

Appendix A. Basic of Differential Geometry

1. Expression of Curves and Surfaces

There are three ways to express a curve in the xy plane — explicit form like $y = f(x)$; implicit form like $F(x, y) = 0$; and parametric form like $p(t) = (x(t), y(t))$. Each form has its own advantages to represent a curve. Surfaces can also be expressed in explicit form $z = f(x, y)$; implicit form $F(x, y, z) = 0$; and parametric form $S(u, v) = (x(u,v), y(u,v), z(u,v))$. In computer graphics curves and surfaces are generally expressed in the parametric form. In the following sections I will introduce some basic ideas of differential geometry of curves and surfaces. For more details about differential geometry please refer to [Do Carmo 1976], [Beach 91].

2. The Differential Geometry of Curves

2.1 Curvature and Torsion

Consider a parametric space curve with the form $\mathbf{P} = \mathbf{P}(x(v), y(v), z(v))$, suppose s is the arc length at point v we have $s = s(v)$. Since $s(v)$ is well defined, monotonic, and has an inverse, the curve $\mathbf{P}$ can be expressed as the function of s: $\mathbf{P} = \mathbf{P}(s)$.

Let $\mathbf{t}(s) = \frac{d\mathbf{P}}{ds} / \left|\frac{d\mathbf{P}}{ds}\right|$, so $\mathbf{t}(s)$ is the unit tangent vector, that is $\mathbf{t}\cdot\mathbf{t} = 1$. Differentiating $\mathbf{t}\cdot\mathbf{t} = 1$ with respect to s gives $\frac{d\mathbf{t}}{ds}\cdot\mathbf{t} = 0$. The $\frac{d\mathbf{t}}{ds}$ is either zero or perpendicular to $\mathbf{t}$, or $\mathbf{t}$ is $\mathbf{0}$. We usually ignore $\mathbf{t} = \mathbf{0}$ and have

$$\frac{d\mathbf{t}}{ds} = k\mathbf{n} \qquad k \neq 0 .$$

Where k is a scalar variable called *curvature* and n is a unit vector perpendicular to t. See Figure A.1

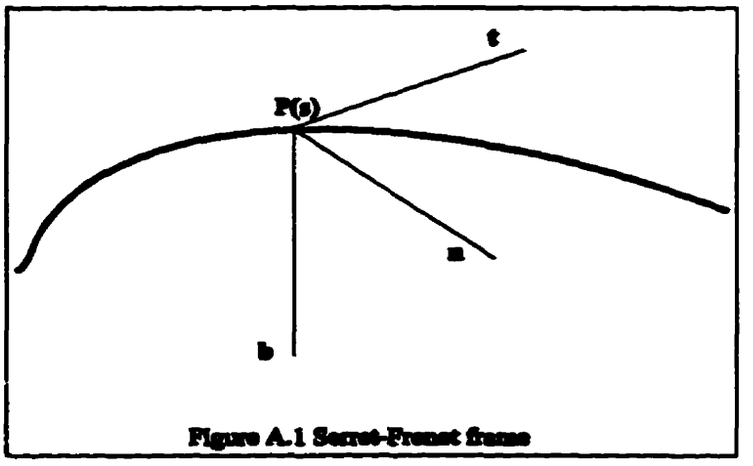


Figure A.1 Serret-Frenet frame

Define a vector $\mathbf{b} = \mathbf{t} \times \mathbf{n}$ then the Serret-Frenet formulas [Do Carmo 1976] are

$$\frac{d\mathbf{t}}{ds} = k\mathbf{n}\,,\ \frac{d\mathbf{n}}{ds} = \tau\mathbf{b} - k\mathbf{t}\,,\ \frac{d\mathbf{b}}{ds} = -\tau\mathbf{n}$$

where n is the principal normal vector, t is the tangent vector, b is the binormal vector, k is the curvature scalar, τ is the torsion scalar, tb plane is called rectifying plane, tn plane is called osculating plane, and nb plane is called normal plane.

Serret-Frenet formulas imply that k(s) and τ(s) determine the space curve except for its position and orientation.

2.2 Osculating Circle

The plane that is determined by vector t and b is known as osculating plane. Consider the circle through P(s) and P(s+Δs) with tangent t at P(s). See Figure A.2. The limiting circle ($\Delta s \to 0$) is called osculating circle. It can be proved that the radius R of the osculating circle is reciprocal the magnitude of the curvature k, that is $R = 1/k$, $k \neq 0$.

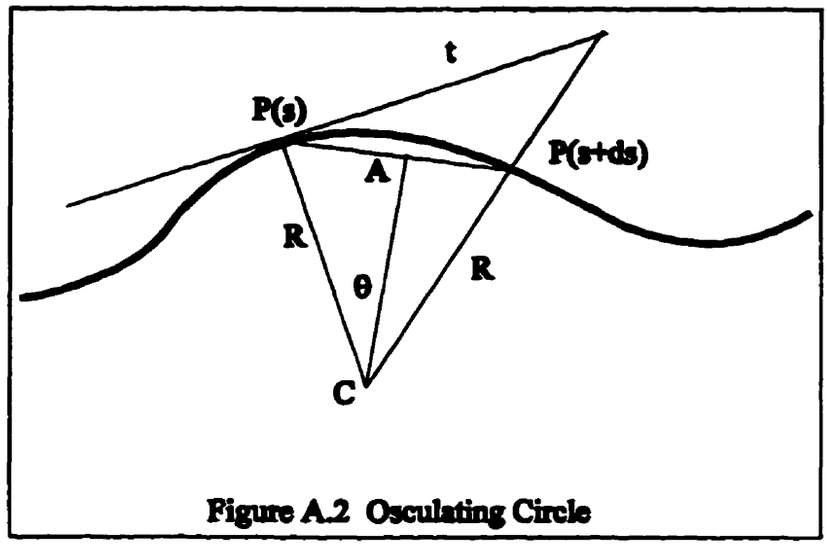


Figure A.2 Osculating Circle

From the Figure A.2 triangles Δ(C, A, P(s))and Δ(C, A, P(s+ds)) are considered congruent for a very small ds. Hence $R*\sin(\theta) = |P(s+ds)-P(s)|/2$. Let $dP(s) = P(s+ds)-P(s)$, by dot product definition, $t \cdot dP(s) = |dP(s)|\cos(\theta)$. Square both sides we get $(t \cdot dP(s))^2 = |dP(s)|^2\cos^2(\theta) = |dP(s)|^2(1 - \sin^2(\theta)) = |dP(s)|^2(1 - |dP(s)|^2/(4*R^2))$. So

$$\frac{1}{R^2} = \frac{4}{|dP(s)|^2}\left(1 - \frac{(t \cdot dP(s))^2}{|dP(s)|^2}\right)$$

When $ds \to 0$, $k^2 = 1/R^2$.

3. The Differential Geometry of Surface

3.1 Parametric Surface

Suppose a parametric surface is $\mathbf{P} = \mathbf{P}(u, v)$ where u, v are parameters of the surface.

The vector $\mathbf{n} = \frac{\partial \mathbf{P}}{\partial u} \times \frac{\partial \mathbf{P}}{\partial v} \Big/ \left| \frac{\partial \mathbf{P}}{\partial u} \times \frac{\partial \mathbf{P}}{\partial v} \right|$ is the unit normal at the point P(u, v).

The differential form is $\frac{d\mathbf{P}}{dt} = \frac{\partial \mathbf{P}}{\partial u}\frac{du}{dt} + \frac{\partial \mathbf{P}}{\partial v}\frac{dv}{dt}$. Let

$$\boxed{E = \frac{\partial \mathbf{P}}{\partial u} \cdot \frac{\partial \mathbf{P}}{\partial u},\ F = \frac{\partial \mathbf{P}}{\partial u} \cdot \frac{\partial \mathbf{P}}{\partial v},\ G = \frac{\partial \mathbf{P}}{\partial v} \cdot \frac{\partial \mathbf{P}}{\partial v}},$$

$$\mathbf{H} = \begin{pmatrix} E & F \\ F & G \end{pmatrix},$$

$$\boxed{\frac{d\mathbf{P}}{dt} \cdot \frac{d\mathbf{P}}{dt} = \frac{d\mathbf{P}}{ds} \cdot \frac{d\mathbf{P}}{ds}\left(\frac{ds}{dt}\right)^2 = \left|\frac{d\mathbf{P}}{ds}\right|^2 \left(\frac{ds}{dt}\right)^2 = \left(\frac{ds}{dt}\right)^2},$$

then

$$\boxed{\left(\frac{ds}{dt}\right)^2 = \begin{pmatrix} \frac{du}{dt} & \frac{dv}{dt} \end{pmatrix} \mathbf{H} \begin{pmatrix} \frac{du}{dt} \\ \frac{dv}{dt} \end{pmatrix}}$$

The right side of the equation is known as the *first fundamental form*, and the matrix H is called the *first fundamental matrix*.

$$\det(\mathbf{H}) = EG - F^2 = \left| \frac{\partial \mathbf{P}}{\partial u} \times \frac{\partial \mathbf{P}}{\partial v} \right|^2 > 0 \quad \text{(by Lagrange's identity)}$$

Now let

$$e=\frac{\partial^2 \mathbf{P}}{\partial u^2}\cdot\mathbf{n} \qquad f=\frac{\partial^2 \mathbf{P}}{\partial u \partial v}\cdot\mathbf{n} \qquad g=\frac{\partial^2 \mathbf{P}}{\partial v^2}\cdot\mathbf{n}$$

and

$$\mathbf{K}=\begin{pmatrix} e & f \\ f & g \end{pmatrix}$$

then

$$\mathbf{k}\left(\frac{ds}{dt}\right)^2\cdot\mathbf{n}=\begin{pmatrix}\frac{du}{dt} & \frac{dv}{dt}\end{pmatrix}\mathbf{K}\begin{pmatrix}\frac{du}{dt} \\ \frac{dv}{dt}\end{pmatrix}$$

The right side of the equation is known as the *second fundamental form* and the matrix **K** is called the *second fundamental matrix.*

3.2 Surface Curvature

There are two ways to define the curvature of surfaces. One uses the parametric representation, the other uses the spherical image. In the parametric expression the curvature is defined by the normal and the derivatives at a point.

3.2.1 Curvature of Parametric Surfaces

Given the following matrix

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

where

$$\begin{aligned} a_{11} &= \frac{fF - eG}{EG - F^2} & a_{12} &= \frac{eF - fE}{EG - F^2} \\ a_{21} &= \frac{gF - fG}{EG - F^2} & a_{22} &= \frac{fF - gE}{EG - F^2} \end{aligned}$$

The eigenvalues of the matrix (a_{ij}) are the principal curvatures k_1 and k_2 at the point p.

The Gaussian curvature κ_g and Mean curvature κ_a are the determinant and 1/2 the trace of the matrix (a_{ij}). Hence

$$k_g = \frac{eg - f^2}{EG - F^2} = k_1 * k_2$$

$$k_a = \frac{Eg - 2Ff + Ge}{2(EG - F^2)} = \frac{k_1 + k_2}{2}$$

k_g is an important index to show the feature of a point on a surface. A point P of a surface is called elliptic if $k_g > 0$, hyperbolic if $k_g < 0$, parabolic if $k_g = 0$ and one of (k_1 , k_2) is not zero, or planar if $k_1 = k_2 = 0$. This classification is referred as the Dupin indicatrix and it does not dependent on the orientation.

3.2.2 Geometrical Representation of Curvature

From the books of [Do Carmo 76] and [Gray 93] the Gauss map is defined as follows. Let **x** be a patch on the surface S. Then the unit normal **n** to **x** viewed as a mapping from S to the unit sphere is called the *Gauss map (spherical image)* of **x**. In other words, to each point p on a patch the Gauss map assigns the point on the unit sphere such that the normal at the point on the unit sphere is parallel to the unit normal at p. It can be proved that the Gaussian curvature $\kappa_g(p)$ at the point p of the surface S to be

$$\kappa_g(p) = \lim_{A \to 0} \frac{A'}{A}$$

where A is the area containing point p, A' is the area of the Gauss map A on unit sphere, and the limit is taken through a sequence of nested A that converges to p.

We can classify the points on a surface into elliptic, hyperbolic, parabolic, and planar by the spherical image of Gaussian curvature. When we move a small closed curve without self-intersection points around a point on the surface, its spherical image is also a closed curve without self-intersection points on the sphere. For a elliptic point when we traversed the closed curve in a clockwise direction, the corresponding curve on the spherical image is traversed in a clockwise direction; for hyperbolic point the corresponding curve on the spherical image is traversed in a counter-clockwise direction. As you move around a planar point, its spherical image is a point; as you move around a parabolic point, its spherical image is a curve. Refer to Figure A.3.

Appendix A. Basics of Differential Geometry

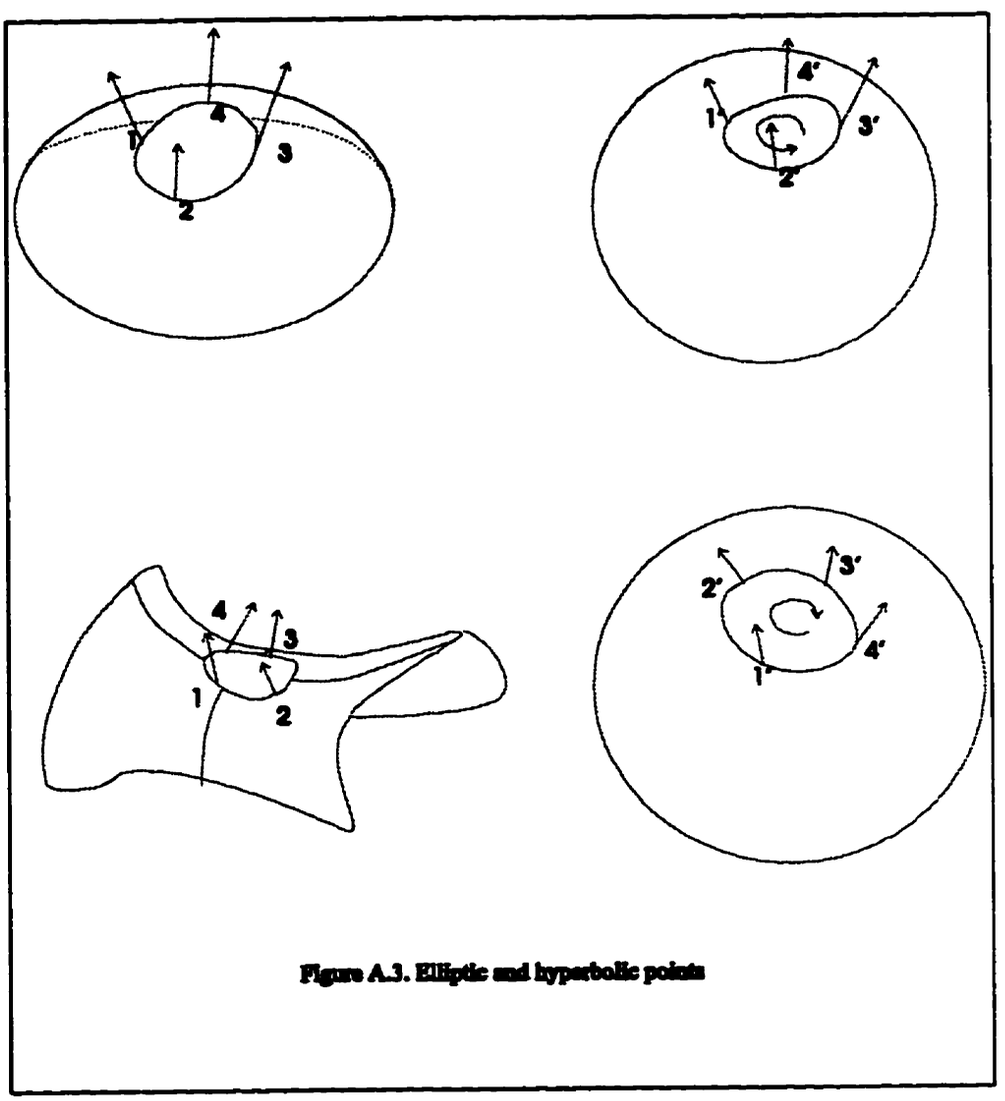


Figure A.3. Elliptic and hyperbolic points

Bibliography

[Averbuch et al 96] A. Averbuch, D. Lazar, and M. Israeli "Image Compression Using Wavelet Transform and Multiresolution Decomposition", IEEE Trans. On Image Processing, Vol.15, No. 1, Jun. 1996

[Beach 91] R. C. Beach "An Introduction to the Curves and Surfaces of Computer-Aided Design" Van Nostrand Reinhold, New York, 1991

[Do Carmo 76] M.P., Do Carmo "Differential Geometry of Curves and Surfaces", 1976, Prentice-Hall, Inc., Englewood Cliffs, New Jersey

[Chui 92] C. K. Chui. "An Introduction to Wavelets". Academic Press, Boston 1992

[Chui & Quak 92] C. K. Chui. and E. Quak "Wavelet on a bounded interval", Numerical Methods in Approximation Theory, In D.Braess and L.L. Schumaker, editors, Vol. 9, 53-75, 1992. Birkhauser Verlag

[Cohen et al 80] E. Cohen, T. Lynche, and R.F. Riesenfeld "Discrete B-Spline and subdivision Techniques in Computer Aided Geometric Design and Computer Graphics", Computer Graphics and Image Proc., Vol. 14, 87-111, 1980

[CRC 88] CRC Standard Math Tables. 28th Edition, CRC Press, 122, 1988.

[de Boor 72] C., de Boor "On Calculation with B-splines", Journal of Approximation Theory, Vol. 6, 50-62, 1972

[Daubechies 88] I. Daubechies "Orthonormal bases of compactly supported wavelets", Communication of Pure Application Mathematics, Vol 41, 909-996, 1988

Bibliography

[DeVore et al 92] R.. DeVore, B. Jawerth, and B. J. Lucier "Image Compression Through Wavelet Transform Coding", IEEE Trans. On Info. Theory Vol.38, No.2, 1992, Special Issue on Wavelet Transform and Multiresolution Signal Analysis

[Eck et al 95] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, W. Suetzle "Multiresolution Analysis of Arbitrary Meshes", Technical Report #95-01-02, University of Washington, Seattle, WA

[Farin 93] G. Farin, "Curves and Surfaces for Computer-Aided Geometric Design, A Practical Guide" Academic Press, New York, 1993

[Farin 95] G. Farin, "NURB Curves and Surfaces: from projective geometry to practical use" A. K.Peters, Ltd. Wellesley, MA 02181, 1995

[Finkelstein & Salesin 94] A. Finkelstein and D. H. Salesin. "Multiresolution curves". In SIGGRAPH 94 Conference Proceedings, Computer Graphics Annual Conference Series, 261-268, July 1994

[Foley & Van Dam 92] J. Foley, and Van Dam, A. "Fundamentals of Interactive Computer Graphics" Addison-Wesley, Reading 1992

[Forsey & Bartels 95] D.R. Forsey, R.H. Bartels "Surface fitting with hierarchical splines", ACM Transactions on Graphics 14(1995), 134-161

[Golub & Van Loan 89] G, H. Golub and C. F. Van Loan. "Matrix Computations", The Johns Hopkins University Press, Baltimore, second edition, 1989

Bibliography

[Gortler etal 93] S. J. Gortler, P. Schroder, M. F. Cohen, and P. Hanrahan. "Wavelet radiosity", In SIGGRAPH 93 Conference Proceedings, Computer Graphics Annual Conference Series, 221-230, August 1993.

[Gray 93] A. Gray "Modern Differential Geometry of Curves and Surfaces", 1993 by CRC Press, Inc.

[Hagen 92] H. Hagen "Surface Interrogation Algorithms", IEEE Computer Graphics and Applications Sept. 12(1992), 53-60

[Hamann 94] B.Hamann "A Data Reduction Scheme for Triangulated Surfaces", Computer-Aided Geometric Design 11, (1994), 197-214

[Hilbert and Cohn-Vossen 52] D.Hilbert and S.Cohn-Vossen "Geometry and the Imagination" translated by P.Nemenyi, 1952 Chelsea Publishing Company

[Koenderink 92] J. Koenderink "Surface Shape and Curvature Scales", Image and Vision Computing 10(1992), 557-565

[Lee et al 93] C-K Lee, R.M. Haralick and K. Deguchi "Estimation of curvature from sampled noisy data", Proc. Computer Vision Pattern Recognition, New York 1993, 536-541

[Lewis & Knowles 92] A.S. Lewis and G. Knowles "Image Compression Using the 2-D Wavelet Transform" Vol. 1, No.2, Apr. 1992, 244-250

[Lounsbery 94] J. M. Lounsbery "Multiresolution Analysis for Surfaces of Arbitrary Topological Type" Ph.D. Thesis, 1994, Dept. of Computer Science and Engineering, University of Washington

Bibliography

[Mallat 89] S. Mallat "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", IEEE Trans. on Pattern Analysis and Machine Intelligence, 11(7):674-693

[Mann & DeRose 95] S. Mann and T. DeRose "Computing values and derivatives of Bezier and B-spline tensor products", Computer Aided Geometric Design Vol. 12, No. 1, Feb. 1995, 105-110

[Meek & Walton 92] D. S. Meek, and D. J. Walton "Approximation of Discrete Data by G^1 Arc Splines" Computer Aided Geometric Design Vol. 24, Num. 6, 301-306,1992

[Pentland 92] A. P. Pentland "Fast Solutions to Physical Equilibrium and Interpolation Problems", Visual Computer, 8 (5-6):303-314, June 1992

[Piegl & Tiller 87] L. Piegl and W. Tiller "Curve and Surface Constructions Using Rational B-Spline" Computer Aided Design, Vol. 19, 485-498, 1987

[Press et al 92] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Fetterling. "Numerical Recipes", Cambridge University Press, second edition, 1992

[Quak and Weyrich 93] E. Quak and N. Weyrich "Decomposition and Reconstruction Algorithms for Spline Wavelets on a Bounded Interval", Technical Report 294, Department of Mathematics, Texas A&M University, April 1993

[Rogers & Adams 90] D. F. Rogers, and J. A. Adams, " Mathematical Elements for ComputerGraphics" McGraw-Hill, Inc. New York, 1990

[Rosenfeld 88] B. A. Rosenfeld "A History of Non-Euclidean Geometry" 1988, Springer-Verlag New York Inc.

Bibliography

[Schroder et al 93] P. Schroder, S.J.Gortler, M.F.Cohen, and P.Hanrahan "Wavelet Projections for Radiosity". In Proceedings of the Fourth Eurographics Workshop on Rendering, Pages 105-114, Paris, France, 1993

[Stollnitz et al 94] E. J. Stollnitz, T. D. DeRose, D. H. Salesin "Wavelets for Computer Graphics: A Primer", Technical Report 94-09-11, Dept. of Computer Science and Engineering, University of Washington, Seattle, Washington 98195, 1994

[Stoddart et al 95] A. J. Stoddart, J. Illingworth and T.Windeatt "Optimal parameter selection for derivative estimation from range images", Image and Vision Computing 13 (1995), 629-635

[Stokely & Wu 92] E.M. Stokely and S.Y. Wu, "Surface Parameterization and Curvature Measurements of Arbitrary 3-D Objects: Five Practical Methods", IEEE Trans. Pattern Analysis Machine Intelligence, Vol. 14, No.8, 833-840, 1992

[Todd 86] P. H. Todd, "Numerical Estimation of the Curvature of Surfaces", Computer-Aided Design 18(1986), 33-37