2008

Multiresolution editing for B-spline curves and surfaces

Bo Xu
Iowa State University

Multiresolution editing for B-spline curves and surfaces

by

Bo Xu

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Industrial Engineering

Program of Study Committee:
James H. Oliver, Major Professor
Adrian Sannier
Carolyn Heising
Matthew Frank
Robyn R. Lutz

Iowa State University

Ames, Iowa

2008

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to those who assisted me with various aspects of conducting this research and the writing of this dissertation. First and foremost, I am grateful to my advisors and mentors, Dr. James H. Oliver and Dr. Adrian Sannier, for their guidance, patience, and support throughout my Ph.D. program of study, this research, and the writing of the dissertation. Their insights and words of encouragement have often inspired me toward the completion of my graduate education.

I would also like to thank my committee members, Drs. Carolyn Heising, Matthew Frank, and Robyn R. Lutz, for their advices and contributions to this work.

I am also grateful to my colleagues and friends—Faneng Sun, Hongtao Qi, Jinran Chen, Liangshou Wu, Junhua Pang, Fujun Wang, Michael Carter, Tsung-Pin Yeh, Jeremy Bennett, Sashank Ganti, Jianbing Huang, and my many friends from VRAC—who supported me along the way with their friendship and shared expertise, which helped to build wonderful memories.

Finally, and most importantly, I am especially thankful to my wife, my parents, and the remainder of my family for their continual love and support. They have made the long journey to complete my Ph.D. study a very enjoyable experience.

ABSTRACT

Since 1980 surface modeling has been used in industrial design, CAD and entertainment to create and represent complex forms. Even with this comparatively long history of development, challenges remain in free-form surface modeling. One such challenge is building surface creation and editing techniques that effectively balance the need for local control with the need to control the overall global shape, or sweep of the surface. This dissertation presents a multiresolution approach to the creation of surfaces that allows a designer to more easily manage this balance between local and global control. The techniques presented in this dissertation utilize a wavelet decomposition of B-spline curves and surfaces to allow a designer to easily develop the basic shape using lower level representations, and then seamlessly switch to higher level representations to achieve fine control over local features. The algorithms described in the dissertation are implemented in an interactive software system that is used to demonstrate their effectiveness in comparison to existing methods.

CHAPTER 1. INTRODUCTION

A necessary feature of a modern CAD system is the facility for free-form surface modeling, supplying users with tools and techniques for interactive free-form surface creation and editing. These surfaces are typically defined by linear combinations of parametric functions, such as B-splines that spatially blend a set of control points. Free-form surface modeling is accomplished by defining and manipulating these sets of control points. Often the interfaces for creating these control point nets clearly reflect the underlying mathematics of the modeling methods. Although such interfaces are the most straightforward to the program and are often the easiest and fastest way from a mathematical point of view, they are not necessarily the most intuitive for a user.

A central tradeoff involved in crafting intuitive surface creation and editing interfaces involves balancing the desire to easily control overall global shape, or sweep of the curve or surface with the need to define localized detail. Since the local property of B-splines limits the balance between local and overall sweep control, free-from surface modeling using spline based surfaces can become extremely tedious and in some cases even impossible.

Multiresolution analysis techniques have recently emerged as a fundamental paradigm to resolve this conflict, primarily because they allow a function to be described in terms of a coarse overall shape augmented by details that range from broad to narrow. This more flexible representation brings with it other useful properties such as progressively refinable reconstruction, efficient curve and surface compression, error bounds, and efficient computational algorithms [1].

This dissertation presents a new multiresolution approach to the creation of certain classes of free-form curves and surfaces that allows a designer to more easily manage the balance between local and global control. The techniques presented in this dissertation utilize a wavelet decomposition of a B-spline curve or surface to allow a designer to easily develop the basic shape, and then seamlessly switch to higher levels of detail to add additional definition. A surfacing kernel based on these wavelet-based B-splines is used as the basis for developing 3D space curve and free-form surface modeling techniques. The algorithms described in the dissertation are implemented in a software system that is used to illustrate their utility and compare them to existing methods.

1.1 Literature Review

B-splines have become the de facto industry standard for geometric modeling, graphics, and graphical information exchange. They provide a unified mathematical representation for free-form curves, surfaces, and standard analytic shapes. The excellent mathematical properties and advanced geometric algorithms of B-splines have contributed to the enormous popularity of their representations in geometric modeling systems [2, 3, 4]. In recent years, much work has been focused on methods that allow the user to effectively manage the balance between local and global control of B-splines.

The transformation deformation technique of Barr [5] was a step in this direction. The transformations Barr defined included stretching, bending, twisting, and tapering operators. The shaping tools were designed by Cobb [6] specifically for B-splines surfaces. They use a method to move control points in groups, and then apply linear and nonlinear transformations to them. While their approach does improve the user’s ability to control the global shape of the surface by allowing the movement of many control points at one time, arbitrary deformations are not possible. It is often unclear which control points should be moved to achieve particular effects or how transformations will affect the object.

The free-form deformation (FFD) for global editing, developed by Sederberg and Parry [7], represented an important step forward, allowing objects to be deformed in a more controlled and meaningful way. In the FFD method, the geometric model is embedded in a parallelepiped lattice of control points defining a volume in which the surface control points are embedded. The deformations of the volume lattice are then mapped to corresponding deformations of the embedded surface control points, thus affecting smooth global shape manipulation of the surface. Coquillart [8] later extended this method to include non-parallelepiped lattices. Instead of starting with the FFD's parallelepiped lattice, the user defines the shape of the initial lattice to induce the intended deformation. The extended FFD overcame some construction and deformation limitations of the original FFD. However, the user must know the deformation shape before starting to model, and the interface still directly reflects the underlying mathematics of the modeling method.

Hierarchical B-splines (HBS) as a basis for multiresolution modeling were presented in original work on hierarchical editing by Forsey and Bartels [9]. The basic idea behind HBS is to add finer B-spline patches onto an existing coarse B-spline patch to localize the effect of refinement. The finer patches are created by using hierarchically controlled subdivisions, and form an overrepresentation for the original surface. Forsey and Bartels observed that the refinement frames of reference fixed upon the surface being edited are essential to achieve correct editing. Using this method, one can build very complicated shapes; however the method does not provide a rigorous analysis framework. Forsey and Wong [10] extended

HBS to an analysis algorithm using weighted least squares, but it remains too computationally expensive to run interactively.

Hierarchical triangle splines [11] extended HBS to smooth surfaces of arbitrary topology. Based on the triangular interpolation scheme [12], this method enables hierarchical editing by interpolating the vertices of a hierarchy of locally refined meshes. Each local surface refinement replaces a set of coarse surface patches by a set of finer surface patches while maintaining both overall tangent continuity and geometric shape. This method can deal with any triangular mesh without restrictions of topology and geometry. Since the user decides where to add a finer surface patch, the overrepresentation is not unique. The choices a user makes in adding these refinement areas introduce significant constraints on further model development. Subsequent editing of a model is strongly dependent on the model's editing history, and this history can grow quite complex. The dependence on editing history runs contrary to the desire to be able to easily edit existing models.

Wavelets provide a framework to rigorously define a multiresolution analysis, akin to those defined by HBS, but with a fast, general analysis algorithm. Finkelstein and Salesin [13] used a wavelet based decomposition to describe a multiresolution B-spline curve representation. Their representation does support interactive curve editing, such as overall shape editing, fractional-level editing and detail editing. In their method, detail editing is accomplished by replacing one set of detail coefficients with a new set, allowing a user to edit the detailed characteristics of the curve without affecting its overall sweep. Since detail maintains its orientation as the sweep is changed, a local parameterization of detail can be used to orient detail with respect to coarser level tangent/normal frames. While this detail editing can be performed interactively, it does not support direct manipulation of the curve's detailed shape as a designer would like, and as such is unsuitable as a general purpose shape editing tool. Gortler and Cohen [14] point out that directly manipulating wavelet representations of detail does not produce an intuitive interface, due to the non-intuitive shape of the underlying wavelet functions. Instead of the smooth shape changes a modeler would like to see in geometric space, manipulations of the wavelet coefficients results in shape changes with “extra wiggles”.

Lousbery [15] set up the connection between wavelets and subdivision, and used it to define multiple levels of resolution. The original constructions were directly applied to piecewise linear subdivision, but smoother constructions have also been identified [16, 17, 18]. Lousbery explored the possibility of multiresolution editing based on subdivision wavelets. This kind of editing clearly reflects the underlying mathematics of the modeling method, but it could not provide the user with an intuitive and interactive interface. The multiresolution representation is restricted to polygonal meshes with subdivision connectivity at the finest level. Eck [19] presented a method for overcoming this shortcoming and could convert any arbitrary mesh into multiresolution form.

Constraint-based multiresolution editing techniques have proved to be helpful for geometric modeling [20, 21]. Elber [22] developed a scheme that incorporates linear constraints into a multiresolution editing environment, allowing the user to perform multiresolution editing for non-uniform B-spline curves, while satisfying positional, tangential, and orthogonal constraints on the curves. He also showed that area preservation can be viewed as a linear constraint and can be reformulated into the multiresolution framework. Hahmann and Sauvage [23] explored the area preservation problem in a rigorous analysis framework based on wavelets. In their method, area preservation is expressed as a bilinear form of the coarse control points and wavelet coefficients through all levels of resolution, and multiresolution deformation with the area constraint is computed through an optimization process. To enable the efficient computation of the area in the wavelet decomposition form, they built their multiresolution editing framework based on uniform B-splines. Sauvage [24] generalized B-spline area preservation to three-dimensional B-spline surfaces with volume preservation. Multiresolution deformations for both uniform and non-uniform B-splines are addressed in his framework. Although a minimization process is required to develop the solutions, this method can support the interactive detail-preserving editing of B-spline surfaces with volume constraint. Another constraint of high interest is arc-length preservation of the curves. Sauvage and Hahmann [25] described arc-length preservation as a non-linear constraint, and integrated it into a multiresolution editing system. They later extended it into a wrinkling tool for soft tissue deformation [26]. Obviously, constraint-based multiresolution editing techniques can offer additional and finer control over the deformations of the curves and surfaces. However, since the deformation at a coarse level is computed from its underlying optimization mathematics, the ambiguity of the deformation can not provide the user with an intuitive design view.

Previous work has shown that multiresolution techniques can help balance the need for targeted, local control of surface detail with the need to control the overall shape of the surface. While wavelets have been considered as a basis for multiresolution modeling, a suitable method has so far not been identified. The contribution of the research presented in this dissertation is to demonstrate an effective method for applying a wavelet based multiresolution analysis as the basis for an interactive curve and surface creation and editing mechanism.

1.2 Dissertation Overview

This dissertation presents a multiresolution approach to the creation of certain classes of free-form curves and surfaces that allows a designer to more easily manage the balance between local and global control.

1.2.1 Outline

Before going into the details of multiresolution analysis for B-splines, it is first necessary to present background information. Chapter 2 introduces the mathematical framework of multiresolution analysis and then discusses three different classes of wavelet bases: orthogonal wavelets, semiorthogonal wavelets, and biorthogonal wavelets.

Chapter 3 explains how to construct B-spline wavelets, one of the more important wavelets used in hierarchical representation of free-form curves and surfaces.

The primary contribution of this dissertation is presented in Chapter 4, which discusses the details of multiresolution editing techniques for B-spline curves and surfaces. This includes: (1) multiresolution editing for B-spline space curves; (2) multiresolution editing for B-spline surfaces.

Applications of multiresolution surface editing, including product design and styling, are discussed in Chapter 5.

Finally, Chapter 6 summarizes this work and suggests interesting areas of future research in B-spline wavelets, including applications that have not yet been made concrete.

1.2.2 Contributions

The specific contributions for the research include:

- *Designing and developing an intuitive and convenient editing framework for B-spline modeling, which uses the connection between B-spline wavelets and subdivision to define different levels of resolution*. Based on Finkelstein and Salesin's earlier work, our approach overcomes the undesirable detail editing characteristics inherent in their formulation.
- *Extension of this one-dimensional multiresolution editing technique to two-dimensional situations.* This extension introduces a multiresolution editing technique for B-spline surface modeling.
- *Unification of the multiresolution curve and surface editing techniques.* During the modeling process, B-spline geometry construction is broken into several steps. Each of these steps has its own multiresolution representation, enabling design-specific requirements to be met.
- *Implementation of these algorithms in a software system that can provide interactive performance on commodity hardware.* Since wavelet based curves and surfaces can be decomposed and reconstructed in linear time, the design of dynamic data structures and algorithms can make manipulation fast enough to maintain interactivity.

CHAPTER 2. MULTIRESOLUTION ANALYSIS BASED ON WAVELETS

Wavelets are a mathematical tool for hierarchically decomposing functions [27]. Multiresolution analysis based on wavelets can help balance the need for targeted, local control of detail with the need to control the overall shape of the function. In this section, the mathematical theory of wavelets and multiresolution analysis is presented, as well as methods to combine them together to develop a rigorous analysis framework.

2.1 Haar: The Simplest Wavelet Basis

2.1.1 The One-dimensional Haar Wavelet Transform

To illustrate how wavelets work, the simplest wavelets – Haar wavelets are first introduced. Suppose a sequence of numbers is given, having the following values:

$$(7\ \ 5\ \ 2\ \ 6)$$

The sequence can be represented in a wavelet transform using the Haar basis. First, the numbers are averaged in a pairwise manner, and a lower-resolution representation is obtained with the result as follows:

$$(6\ 4)$$

Clearly, this sequence has lost some information from the initial one. To be able to recover the initial sequence, some "detail coefficients" need to be defined and used to capture the missing information. In this example, the first detail coefficient can be taken as 1 and the second one as -2, so the initial sequence can be recovered from the following operations:

$$6 + 1 = 7$$

$$6 - 1 = 5$$

$$4 + (-2) = 2$$

$$4 - (-2) = 6$$

Thus the initial sequence (7 5 2 6) can be represented as (6 4 1 -2) without loss of information. The numbers (6 4) are the coarse values which represent a low-resolution representation, and the numbers (1, -2) are the detail coefficients which capture the missing information. The averaging decomposition can be repeated recursively. The further decomposition from the sequence (6 4 1 -2) yields the new sequence (5 1 1 -2). Now the number 5 is the single value representing the overall average of the original sequence, and the numbers (1 1 -2) are the detail coefficients in order of increasing resolution. The process that repeats the averaging decomposition recursively is called the Haar wavelet decomposition, and new sequence (5 1 1 -2) is the Haar wavelet transform of the initial sequence (7 5 2 6).

2.1.2 One-dimensional Haar Basis Functions

For piecewise constant functions, consider the half-open interval [0, 1). A one number sequence is just a function that is constant over the interval [0, 1). V^0 is defined as the vector space of all such functions. A two number sequence has two constant pieces over the interval [0, 1/2) and [1/2, 1). V^1 is denoted as the space containing all these functions. If this is continued further, the space V^j will include all piecewise constant functions in the interval [0, 1) with constant pieces over each of 2^j subintervals. Note that a piecewise constant function in the space V^j can always be described as a pair of piecewise constant functions in the space V^{j+1}. Thus, the vector spaces V^j are nested, and can be expressed as:

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

The basis functions of the vector spaces V^j are called scaling functions. A basis for the vector space V^j is the set of scaled and translated "box" functions:

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

where

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

Figure 2.1 shows the box basis for the space V^2.

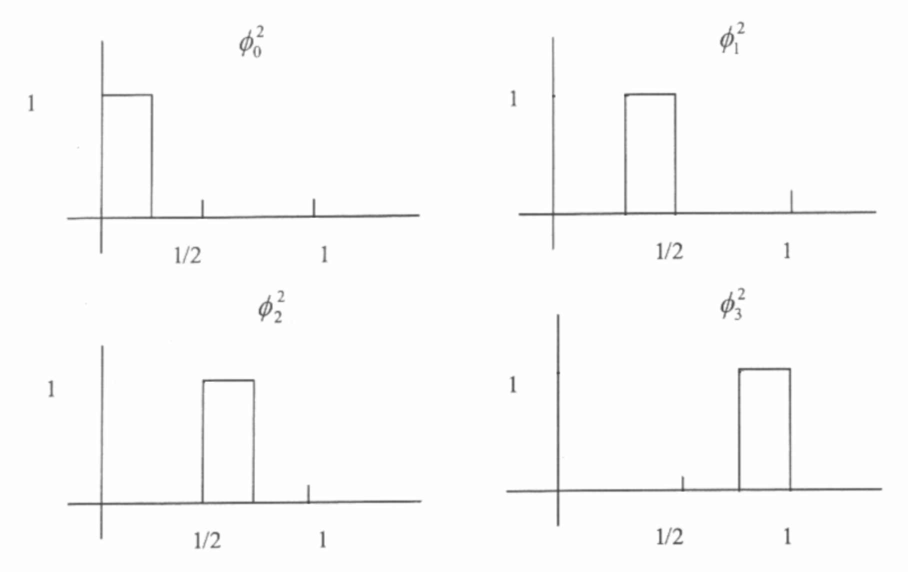


Figure 2.1 The Haar scaling functions for the space V^2.

The next step in building a multiresolution analysis is to define wavelet spaces. A wavelet space W^j is defined as the orthogonal complement of the space V^j in the space V^{j+1}. In other words, the wavelet space W^j is the subspace of the space V^{j+1}, and any function in the space W^j is orthogonal to the functions in the space V^j. The basis functions for the space W^j are called wavelets. The wavelets corresponding to the box scaling functions are known as Haar wavelets, and can be specified as

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

where

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

Figure 2.2 show the two Haar wavelets for the space W^1.

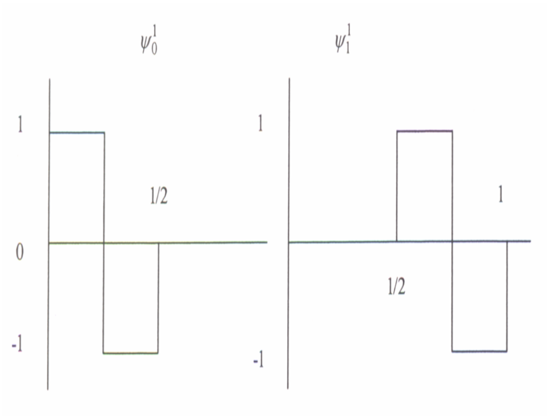


Figure 2.2 The Haar wavelet functions for the space W^1.

Since the wavelet space W^j is the orthogonal complement of the space V^j in the space V^{j+1}, the basis functions of the space W^j, together with the basis functions of the space V^j, form a basis for the space V^{j+1}. Now, these ideas can be applied to the previous example. The initial sequence (7 5 2 6) can be represented as

$$f(x) = 7\phi_0^2(x) + 5\phi_1^2(x) + 2\phi_2^2(x) + 6\phi_3^2(x)$$

It can also be represented in V^1 and W^1 as

$$f(x) = 6\phi_0^1(x) + 4\phi_1^1(x) + 1\psi_0^1(x) + (-2)\psi_1^1(x)$$

Finally, it can be represented in V^0, W^0, and W^1 as

$$f(x) = 5\phi_0^0(x) + 1\psi_0^0(x) + 1\psi_0^1(x) + (-2)\psi_1^1(x)$$

Once again, the sequence (5 1 1 -2) is the Haar wavelet transform of the initial sequence (7 5 2 6). The original function is represented as the overall average combined with the wavelet details.

2.2 Multiresolution Analysis and Wavelets

2.2.1 Multiresolution Analysis

The basic idea behind multiresolution analysis is to decompose a complicated function into a simpler, low resolution function which, when combined with a specific detail representation, can be used to recover the original function [28, 29]. There are two basic ingredients for a multiresolution analysis: an infinite chain of nested vector spaces $V^0 \subset V^1 \subset V^2 \subset \cdots$ and an inner product $\langle f, g \rangle$ defined on any pair of functions $f, g \in V^j$, $j < \infty$ [30]. The basic functions of the space V^j are called scaling functions. The next step in building a multiresolution analysis is to define the wavelet spaces W^j. To envelop a wide variety of wavelet constructions, the wavelet spaces W^j are defined as the complement of the space V^j in the space V^{j+1}. Thus, any function in the space V^{j+1} can be expressed uniquely as the sum of a low resolution function in the space V^j and a detail function in the space W^j. The basis vectors of the wavelet space W^j are called wavelets.

On a bounded domain, the basis of the space V^j is finite, allowing us to use matrix notation, as in Quak and Weyrich [31].

2.2.1.1 Refinement

Representing scaling functions $\Phi^j(x)$ and wavelet functions $\Psi^j(x)$ for a given level j in a single row matrix:

$$\Phi^j(x) = [\phi_0^j(x) \cdots \phi_{m^j-1}^j(x)]$$

$$\Psi^j(x) = [\psi_0^j(x) \cdots \psi_{n^j-1}^j(x)]$$

where m^j is the dimension of the space V^j and n^j the dimension of the space W^j. Because the space W^j is the complement of the space V^j in the space V^{j+1}, the dimensions of these spaces satisfy $m^{j+1} = m^j + n^j$.

Since the vector space V^j is nested, it requires that the scaling functions be refinable. That is, for all $j = 1, 2, \cdots,$ there exists a constant matrix P^j such that

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

where P^j is a $m^j \times m^{j-1}$ matrix.

The wavelet space W^{j-1} is by definition the complement of the space V^{j-1} in the space V^j, so it is also a subspace of the space V^j. The wavelets $\Psi^{j-1}(x)$ can be written as linear combinations of the scaling functions $\Phi^j(x)$. This means there is a $m^j \times n^{j-1}$ constant matrix Q^j satisfying

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

Equations (2.1) and (2.2) can be combined together in block-matrix notation and represented as:

$$\left\lfloor \Phi^{j-1} \middle| \Psi^{j-1} \right\rfloor = \Phi^j \left\lfloor P^j \middle| Q^j \right\rfloor \tag{2.3}$$

2.2.1.2 The filter bank

Consider a function in some vector space V^j, expressed as a linear combination of the scaling function basis $\Phi^j(x)$. The coefficients of the function in terms of the basis can be written as a column matrix of values $C^j = [c_0^j \cdots c_{m^j-1}^j]^T$.

Suppose a low-resolution version C^{j-1} is desired with fewer coefficients m^{j-1} than has C^j. The standard method for creating C^{j-1} is to do some form of linear filtering and down-sampling on the entries of C^j. This process can be expressed as:

$$C^{j-1} = A^j C^j \tag{2.4}$$

where A^j is a $m^{j-1} \times m^j$ constant matrix.

Since C^{j-1} contains fewer coefficients, some amount of detail is lost in this filtering process. The lost detail can be captured by another matrix equation:

$$D^{j-1} = B^j C^j \tag{2.5}$$

where B^j is a $n^{j-1} \times m^j$ constant matrix. The process of splitting the coefficients C^j into a low-resolution version C^{j-1} and detail D^{j-1} is called decomposition. The matrices A^j and B^j are called analysis filters.

The original coefficients C^j can be reconstructed from its low-resolution version C^{j-1} and detail D^{j-1}:

$$C^j = P^j C^{j-1} + Q^j D^{j-1} \tag{2.6}$$

where P^j and Q^j are the same matrices as in equation (2.1) and (2.2), and are called synthesis filters. This process of recovering C^j from C^{j-1} and D^{j-1} is called reconstruction.

The decomposition procedure can be applied recursively to the low-resolution version C^{j-1}. Thus, the original coefficients can be expressed as a hierarchy of lower-resolution versions $C^0, \cdots, C^{j-1}$ and details $D^0, \cdots, D^{j-1}$, as shown in Figure 2.3. The recursive process is known as a filter bank.

Note that the original coefficients C^j can be recovered from the sequence $C^0, D^0, D^1, \cdots, D^{j-1}$. This sequence is called a wavelet transform. Since the total size of the transform $C^0, D^0, D^1, \cdots, D^{j-1}$ is the same as that of original version C^j, no extra storage is required.

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

Figure 2.3 The filter bank.

In general, A^j and B^j are formed by satisfying the relation:

$$\left[\Phi^{j-1} \middle| \Psi^{j-1}\right]\left[\frac{A^j}{B^j}\right] = \Phi^j \qquad (2.7)$$

Combining equations (2.3) and (2.7) gives

$$\left[\frac{A^j}{B^j}\right] = \left[P^j \mid Q^j\right]^{-1} \qquad (2.8)$$

where $\left[\frac{A^j}{B^j}\right]$ and $\left[P^j \mid Q^j\right]^{-1}$ are both square matrices.

2.2.2 Orthogonal Wavelet System

An orthogonal wavelet system is defined as one in which the scaling functions are orthogonal to each other, the wavelet functions are orthogonal each other, and each of the wavelets is orthogonal to every scaling functions at the coarser level.

The mathematical representation of orthogonality can be written as:

$$\begin{aligned} &\left\langle \phi_k^j \middle| \phi_l^j \right\rangle = \delta_{k,l} \\ &\left\langle \psi_k^j \middle| \psi_l^j \right\rangle = \delta_{k,l} \quad \textit{for all } j, k, \text{and } l \\ &\left\langle \phi_k^j \middle| \psi_l^j \right\rangle = 0 \end{aligned} \tag{2.9}$$

where $\delta_{k,l}$ is defined to be 1 if $k = l$, and 0 otherwise.

Consider two row matrix functions $\Phi(x) = [\phi_0(x)\,\phi_1(x)\cdots]$ and $\Psi(x) = [\psi_0(x)\,\psi_1(x)\cdots]$. Let $[\langle \Phi | \Psi \rangle]$ denote the matrix whose (k,l) entry is $\langle \varphi_k | \psi_l \rangle$. The following relations exist between these two functions and a constant matrix C:

$$\begin{aligned} [\langle \Phi | \Psi \rangle] &= [\langle \Psi \mid \Phi \rangle]^T \\ [\langle \Phi \mid \Psi C \rangle] &= [\langle \Phi \mid \Psi \rangle] C \\ [\langle \Phi C \mid \Psi \rangle] &= C^T [\langle \Phi \mid \Psi \rangle] \end{aligned} \tag{2.10}$$

The orthogonality conditions in equation (2.9) can be rewritten as:

$$\begin{aligned} [\left\langle \Phi^j \middle| \Phi^j \right\rangle] &= I \\ [\left\langle \Psi^j \middle| \Psi^j \right\rangle] &= I \\ [\left\langle \Phi^j \middle| \Psi^j \right\rangle] &= 0 \end{aligned}$$

where I denotes the identity matrix, and 0 is the zero matrix.

To see what orthogonality implies about the matrices P^j and Q^j, combine the basis functions to produce:

$$[\langle[\Phi^{j-1} \mid \Psi^{j-1}] \,|\, [\Phi^{j-1} \mid \Psi^{j-1}]\rangle] = \begin{bmatrix} [\langle \Phi^{j-1} \mid \Phi^{j-1} \rangle] & [\langle \Phi^{j-1} \mid \Psi^{j-1} \rangle] \\ [\langle \Psi^{j-1} \mid \Phi^{j-1} \rangle] & [\langle \Psi^{j-1} \mid \Psi^{j-1} \rangle] \end{bmatrix} = I$$

Substituting equation (2.3) into the previous equation, leads to:

$$[\langle \Phi^{j}[P^{j} \mid Q^{j}] \,|\, \Phi^{j}[P^{j} \mid Q^{j}] \rangle] = I$$

Considering the relations listed in equation (2.10), one can obtain

$$[P^{j} \mid Q^{j}]^{T} \, [\langle \Phi^{j} \,|\, \Phi^{j} \rangle][P^{j} \mid Q^{j}] = I$$

Since $[\langle \Phi^{j} \,|\, \Phi^{j} \rangle] = I$, the previous equation can be rewritten as:

$$[P^{j} \mid Q^{j}]^{T} = [P^{j} \mid Q^{j}]^{-1}$$

Thus, the matrix $[P^{j} \mid Q^{j}]$ is an orthogonal matrix, which, in combination with equation (2.8), indicates that:

$$A^{j} = (P^{j})^{T} \text{ and } B^{j} = (Q^{j})^{T}$$

Thus, Haar wavelets are a simple case of orthogonal wavelets. It was originally believed that a smooth orthogonal wavelet system did not exist until Daubechies developed the first orthonormal, smooth, and compactly supported wavelet system [32]. The drawback of Daubechies wavelets is that the recursive rather than analytical representation of the wavelet functions limits their practical use.

2.2.3 Semiorthogonal Wavelet System

Orthogonality is not the only desirable property when constructing a wavelet basis. In fact, if smooth symmetric wavelets with compact supports are desired, orthogonality must be sacrificed. Wavelets, which are orthogonal to the scaling functions but not to each other, are referred to as semiorthogonal wavelets. In mathematical notation, semiorthogonality can be written as:

$$\langle \phi_k^j \mid \psi_l^j \rangle = 0 \quad for\ all\ j, k, \text{and}\ l \tag{2.11}$$

The matrix representation of semiorthogonality is:

$$[\langle \Phi^j \mid \Psi^j \rangle] = 0$$

Changing the superscript to $j-1$ and substituting equations (2.1) and (2.2) into the above equation, yields:

$$(P^j)^T [\langle \Phi^j \mid \Phi^j \rangle] Q^j = 0 \tag{2.12}$$

To find a wavelet refinement matrix Q^j, define $K^j = (P^j)^T [\langle \Phi^j \mid \Phi^j \rangle]$. The set of all possible solutions to equation (2.12) is called the null space of K^j. Since the wavelet refinement matrix Q^j determines the wavelets Ψ^j, a multitude of bases for the null space K^j implies that there are the same number of wavelet bases for a given wavelet space W^j. To determine the wavelets, further constraints need to be imposed. B-spline wavelets are a good example of such a semiorthogonal construction. These wavelets are built from B-splines, and have been developed to a large extent by Chui and his colleagues [33, 34]. A detailed description of B-spline wavelets is introduced in the next chapter.

2.2.4 Biorthogonal Wavelet System

Before defining biorthogonal wavelets, dual basis functions must first be introduced. Consider a basis $u(x) = [u_1(x)\ u_2(x)\ \cdots]$ and a collection of functions

$\bar{u}(x) = [\bar{u}_1(x)\ \bar{u}_2(x)\ \cdots]$. The dual basis corresponding to the basis $u(x)$ is the set of functions $\bar{u}(x)$ satisfying the following:

$$[\langle u \mid \bar{u} \rangle] = I$$

Dual basis functions are central to construct biorthogonal wavelets. Suppose $\bar{\Phi}^j(x)$ and $\bar{\Psi}^j(x)$ are the duals corresponding to $\Phi^j(x)$ and $\Psi^j(x)$ respectively. Thus,

$$\begin{aligned} [\langle \Phi^j \mid \bar{\Phi}^j \rangle] &= I \\ [\langle \Psi^j \mid \bar{\Psi}^j \rangle] &= I \end{aligned} \tag{2.13}$$

A biorthogonal wavelet system is one in which the scaling functions are orthogonal to the dual wavelets and the wavelets are orthogonal to the dual scaling functions. In mathematical notation, the conditions defining biorthogonal wavelets can be written as

$$\left.\begin{aligned} \langle \phi_k^j \mid \bar{\psi}_l^j \rangle &= 0 \\ \langle \psi_k^j \mid \bar{\phi}_l^j \rangle &= 0 \end{aligned}\right\} \ \textit{for all}\ j, k, \textit{and}\ l$$

The matrix representation of the conditions is

$$\begin{aligned} [\langle \Phi^j \mid \bar{\Psi}^j \rangle] &= 0 \\ [\langle \Psi^j \mid \bar{\Phi}^j \rangle] &= 0 \end{aligned} \tag{2.14}$$

Combining equations (2.13) and (2.14), a concise statement of biorthogonality is obtained:

$$[\langle [\Phi^j \mid \Psi^j] \,\big|\, [\bar{\Phi}^j \mid \bar{\Psi}^j] \rangle] = \begin{bmatrix} [\langle \Phi^j \mid \bar{\Phi}^j \rangle] & [\langle \Phi^j \mid \bar{\Psi}^j \rangle] \\ [\langle \Psi^j \mid \bar{\Phi}^j \rangle] & [\langle \Psi^j \mid \bar{\Psi}^j \rangle] \end{bmatrix} = I \tag{2.15}$$

Suppose the synthesis matrices P^j and Q^j are used to define scaling functions and wavelets:

$$\left\lfloor \Phi^{j-1} \middle| \Psi^{j-1} \right\rfloor = \Phi^{j} \left\lfloor P^{j} \middle| Q^{j} \right\rfloor \tag{2.16}$$

The dual matrices $\overline{P}^j$ and $\overline{Q}^j$ define the dual scaling functions and wavelets:

$$\left[\overline{\Phi}^{j-1} \middle| \overline{\Psi}^{j-1} \right] = \overline{\Phi}^{j} \left[\overline{P}^{j} \middle| \overline{Q}^{j} \right] \tag{2.17}$$

To determine the dual matrices $\overline{P}^j$ and $\overline{Q}^j$, rewrite the biorthogonality conditions in equation (2.15) using superscript $j-1$:

$$[\left\langle [\Phi^{j-1} \mid \Psi^{j-1}] \middle| [\overline{\Phi}^{j-1} \mid \overline{\Psi}^{j-1}] \right\rangle] = I$$

Next, substitute equations (2.16) and (2.17) into the previous biorthogonality conditions to obtain:

$$[P^j \mid Q^j]^T [\left\langle \Phi^j \middle| \overline{\Phi}^j \right\rangle]\ [\overline{P}^j \mid \overline{Q}^j] = I$$

Since $[\left\langle \Phi^j \middle| \overline{\Phi}^j \right\rangle] = I$, the previous equation can be expressed as:

$$[\overline{P}^j \mid \overline{Q}^j]^T [P^j \mid Q^j] = I \tag{2.18}$$

Recall the relation between the analysis and synthesis matrices in equation (2.8):

$$\left[\frac{A^j}{B^j} \right] = \left[P^j \mid Q^j \right]^{-1}$$

Comparing these equations, leads to:

$$\overline{P}^j = (A^j)^T \tag{2.19}$$

$$\overline{Q}^j = (B^j)^T \tag{2.20}$$

This indicates that a new subdivision scheme is defined by the analysis matrices A^j and B^j. The scaling functions and wavelets of the new scheme are dual to the scaling functions and wavelets of the original scheme. Single-knot wavelets are an example of biorthogonal wavelets, as was recognized by Sweldens [35] as a special case of lifting. Lifting is an operation that transforms a biorthogonal scheme defined by matrices P^j, Q^j, A^j, and B^j into a new biorthogonal scheme in the following fashion:

$$[P^j_{lift} \mid Q^j_{lift}] = [P^j \mid Q^j - P^j S^j] \qquad (2.21)$$

$$\left[\frac{A^j_{lift}}{B^j_{lift}}\right] = \left[\frac{A^j + S^j B^j}{B^j}\right] \qquad (2.22)$$

Any choice of the matrix S^j will construct a biorthogonal scheme. It is also possible to construct dual lifting in the following fashion (see [35]):

$$[P^j_{\overline{lift}} \mid Q^j_{\overline{lift}}] = [P^j + S^j Q^j \mid Q^j] \qquad (2.23)$$

$$\left[\frac{A^j_{\overline{lift}}}{B^j_{\overline{lift}}}\right] = \left[\frac{A^j}{B^j - A^j S^j}\right] \qquad (2.24)$$

CHAPTER 3. B-SPLINE WAVELETS

B-spline wavelets are a class of semiorthogonal wavelets. These wavelets are constructed from B-splines, and have been developed to a large extent [34]. Endpoint-interpolating B-spline wavelets are one of the important wavelet types used in hierarchical representation of curves and surfaces [36]. The followings explain how endpoint-interpolating B-spline wavelets are constructed.

3.1 B-spline Scaling Functions

The first step in designing B-spline wavelets is to select the scaling functions $\Phi^j(x)$ for a nested set of function spaces. This choice determines the nested spaces V^j and the synthesis matrices P^j. The process starts with the general definition of B-splines, and then specifies how to build uniform endpoint-interpolating B-splines.

Given positive integers d and k $(k \geq d)$, and a non-decreasing sequence of real numbers $u_0, \cdots, u_{k+d+1}$ called knots, for $i = 0, \cdots, k$ and $r = 1, \cdots, d$, the B-spline basis functions of degree d are defined recursively as follows:

$$N_{i,0}(u) = \begin{cases} 1 & if \quad u_i \leq u \leq u_{i+1} \\ 0 & \quad otherwise \end{cases} \tag{3.1}$$

$$N_{i,r}(u) = \frac{u - u_i}{u_{i+r} - u_i} N_{i,r-1}(u) + \frac{u_{i+r+1} - u}{u_{i+r+1} - u_{i+1}} N_{i+1,r-1}(u)$$

(The fractions are taken to be 0 when their denominators are 0)

Endpoint-interpolating B-splines of degree d are obtained when the first and last knots have multiplicity $d+1$. The functions $N_{0,d},\cdots,N_{k,d}$ form a basis for the space of piecewise polynomials of degree d with $d-1$ continuous derivatives.

A B-spline curve of degree d is defined by

$$f(u)=\sum_{i=0}^{k} N_{i,d}(u)\cdot c_i \tag{3.2}$$

where the $\{\, c_i \,\}$ are the control points, and the $\{\, N_{i,d}(u) \,\}$ are the B-spline basis functions of degree d on the nonperiodic knot vector $U=\{\underbrace{0,\cdots,0}_{d+1},u_{d+1},\cdots,u_k,\underbrace{1,\cdots,1}_{d+1}\}$.

B-spline curves exhibit a number of properties that provide desirable geometric characteristics for modeling applications, namely:

- *Endpoint interpolation.* When the first and last knots have multiplicity $d+1$, the B-spline curve will pass through the starting and ending control points.
- *Local support property.* For a B-spline curve of degree d, manipulating a control point c_i, the curve changes only in the interval $[u_i,u_{i+d+1}]$; the curve is left unchanged elsewhere. This property enables a designer to modify the curve at selected intervals without influencing the overall sweep of the curve.
- *Strong convex hull property.* A B-spline curve is contained in the convex hull of its control polygon. In fact, each segment is in the convex hull of the $d+1$ control points, and therefore the convex hull affects the shape of the segment.
- *Differentiability.* A B-spline curve is infinitely differentiable in the interior intervals, and it is at least $d-k$ times continuously differentiable at a knot of multiplicity k.

To construct uniformly spaced B-splines specify $k = 2^j + d - 1$ and the equally distributed knot sequence $u_{d+1}, \cdots, u_k$ are chosen to produce 2^j equally spaced interior intervals. This gives $2^j + d$ B-spline basis functions of degree d and level j. These basis functions form a basis for the vector space $V^j(d)$. These functions may be used as the endpoint interpolating B-spline scaling functions. Figure 3.1 shows an example of these functions at level $j = 2$ for the cubic case.

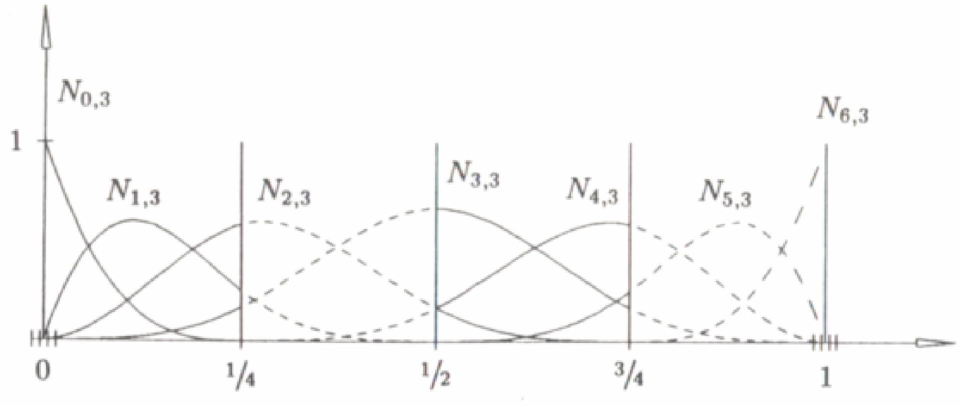


Figure 3.1 Endpoint-interpolating B-spline scaling functions for $V^2(3)$.

From the relationship $\dim(V^j(d)) < \dim(V^{j+1}(d))$ $(j = 0, 1, \cdots)$, any scaling function in the vector space $V^j(d)$ can be expressed as a linear combination of the scaling functions in the vector space $V^{j+1}(d)$. Thus the spaces $V^0(d), V^1(d), \cdots$ are nested as required by multiresolution analysis.

The condition that the vector space $V^j(d)$ is nested is equivalent to requiring that the scaling functions be refinable. That is, for all $j = 1, 2, \cdots$, there must exist a constant matrix P^j (this is the synthesis filter described in equation 2.1) such that:

$$\Phi^{j-1}(x) = \Phi^{j}(x)P^{j} \tag{3.3}$$

where $\Phi^{j}(x) = \left[\phi_0^j(x) \quad \phi_1^j(x) \quad \cdots \quad \phi_{2^j+d-1}^j(x)\right]$.

The synthesis filter P^j is a $(2^j + d)\times(2^{j-1} + d)$ matrix. The entries of the synthesis filter can be developed using the theory of knot insertion for B-splines [31]. Figure 3.2 shows examples of the matrices P^j in case of cubic B-splines. The columns of P^j are sparse, and the interior columns, for $j \geq 3$, are given by vertical translations of the fourth column, shifted down by two places for each column.

3.2 Inner Product

The second step in designing B-spline wavelets is the choice of an inner product, and the standard inner product can be used for this purpose:

$$\langle f, g\rangle := \int_0^1 f(x)g(x)dx \tag{3.4}$$

This choice defines orthogonality between the scaling functions $\Phi^j(x)$ and the wavelet functions $\Psi^j(x)$ in the nested spaces V^j.

3.3 B-spline Wavelets

The final step of developing a multiresolution analysis based on B-splines is to find basis functions for the spaces W^j, which are orthogonal complements to the spaces V^j. The wavelet space W^{j-1} is by definition a subspace of the space V^j, so the wavelet functions $\Psi^{j-1}(x)$ of the space W^{j-1} can be expressed as linear combinations of the scaling functions

$$P^1 = \frac{1}{2}\begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

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

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

Figure 3.2 The synthesis filter P^j for cubic B-splines.

$\Phi^{j}(x)$ of the space V^{j}. This means that there is a $(2^{j}+d)\times(2^{j-1})$ constant matrix Q^{j} satisfying

$$\Psi^{j-1}(x)=\Phi^{j}(x)Q^{j} \tag{3.5}$$

where $\Psi^{j-1}(x)=\left[\psi_{0}^{j-1}(x) \quad \psi_{1}^{j-1}(x) \quad \cdots \quad \psi_{2^{j-1}-1}^{j-1}(x)\right]$.

Since all wavelet functions in $\Psi^{j-1}(x)$ are orthogonal to all scaling functions in $\Phi^{j-1}(x)$, we know that $\left\langle\phi_{m}^{j-1} \mid \psi_{n}^{j-1}\right\rangle=0$ for all m and n. To handle all these inner products simultaneously, the notation for a matrix of inner products is used, as defined in section 2.2.2. The orthogonality condition on the wavelets can be rewritten as:

$$[\left\langle\Phi^{j-1} \mid \Psi^{j-1}\right\rangle]=0 \tag{3.6}$$

Substituting Equation 3.5 into Equation 3.6 yields:

$$[\left\langle\Phi^{j-1} \mid \Phi^{j}\right\rangle]Q^{j}=0 \tag{3.7}$$

The set of all possible solutions of Q^{j} is called the null space of $[\left\langle\Phi^{j-1} \mid \Phi^{j}\right\rangle]$. A multitude of bases for the null space implies that there are different wavelet bases for a given wavelet space W^{j-1}. To uniquely determine the Q^{j} matrices, compact support constraints are imposed to the homogeneous system of the linear equations. Figure 3.3 illustrates the structure of the Q^{j} matrices for cubic B-splines. The matrices Q^{j} are used to compute B-spline wavelet functions. Figure 3.4 illustrates endpoint-interpolating B-spline wavelets for the cubic case at level $j=2$.

At this point, all of the steps in developing a multiresolution analysis based on B-splines are defined. To use B-spline wavelets, wavelet decomposition and wavelet reconstruction processes need to be implemented. Wavelet decomposition allows one to decompose a curve

$$Q^1 = \sqrt{7}\begin{bmatrix} 1 \\ -2 \\ 3 \\ -2 \\ 1 \end{bmatrix}$$

$$Q^2 = \sqrt{\frac{315}{31196288}}\begin{bmatrix} 1368 & 0 \\ -2064 & -240 \\ 1793 & 691 \\ -1053 & -1053 \\ 691 & 1793 \\ -240 & -2064 \\ 0 & 1368 \end{bmatrix}$$

$$Q^3 = \begin{bmatrix} 6.311454 & 0 & 0 & 0 \\ -9.189342 & -1.543996 & 0 & 0 \\ 7.334627 & 4.226722 & 0.087556 & 0 \\ -3.514553 & -5.585477 & -0.473604 & -0.000155 \\ 1.271268 & 6.059557 & 1.903267 & 0.019190 \\ -0.259914 & -4.367454 & -4.367454 & -0.259914 \\ 0.019190 & 1.903267 & 6.059557 & 1.271268 \\ -0.000155 & -0.473604 & -5.585477 & -3.514553 \\ 0 & 0.087556 & 4.226722 & 7.334627 \\ 0 & 0 & -1.543996 & -9.189342 \\ 0 & 0 & 0 & 6.311454 \end{bmatrix}$$

Figure 3.3 The matrices Q^j for endpoint-interpolating cubic B-splines.

$$Q^{j\geq 4}=\sqrt{\frac{5\cdot 2^{j}}{675221664}}\left[\begin{array}{ccccc}
25931.200710 & 0 & 0 & 0 & \\
-37755.271723 & -6369.305453 & 0 & 0 & \\
30135.003012 & 17429.266054 & 385.797044 & 0 & \\
-14439.869635 & -23004.252368 & -2086.545605 & -1 & \\
5223.125428 & 24848.487871 & 8349.373420 & 124 & \\
-1067.879425 & -17678.884301 & -18743.473059 & -1677 & \\
78.842887 & 7394.685374 & 24291.795239 & 7904 & \\
-0.635830 & -1561.868558 & -18420.997597 & -18482 & \\
0 & 115.466347 & 7866.732009 & 24264 & \cdots \\
0 & -0.931180 & -1668.615872 & -18482 & \\
0 & 0 & 123.378671 & 7904 & \\
0 & 0 & -0.994989 & -1677 & \\
0 & 0 & 0 & 124 & \\
0 & 0 & 0 & -1 & \\
 & \vdots & & & \ddots
\end{array}\right]$$

Figure 3.3 (continued)

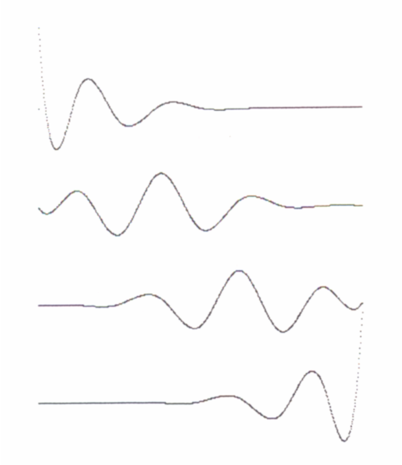


Figure 3.4 Endpoint-interpolating B-spline wavelets for $W^{2}(3)$.

into a low-resolution version and a detail part. Equation 3.8 shows how to compute C^{j-1} and D^{j-1} from C^j in a linear system.

$$[P^j \mid Q^j]\left[\frac{C^{j-1}}{D^{j-1}}\right] = C^j \tag{3.8}$$

where C^j and D^j are column matrices of the corresponding coefficients. The coefficient matrix C^j are typically the x, y, and z coordinates of a curve's control points in $\Re^3$. Wavelet reconstruction can now be used to recover the original curve from its low-resolution and detail parts. The procedure is shown in Equation 3.9.

$$C^j = P^j C^{j-1} + Q^j D^{j-1} \tag{3.9}$$

3.4 Two-dimensional Wavelet Transforms

Wavelet transforms can be extended to two dimensions. In this research, the B-spline wavelet transform is extended to tensor product B-spline surfaces. Thus, multiresolution editing techniques can be applied to 3D tensor product B-spline surfaces.

3.4.1 Tensor Product B-spline Surfaces

A B-spline surface is defined by

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)c_{i,j} \tag{3.10}$$

where the $\{c_{i,j}\}$ is a bidirectional net of control points, and the $\{N_{i,p}(u)N_{j,q}(v)\}$ are the products of the univariate B-spline basis functions on the nonperiodic knot vectors, $U = \{\underbrace{0,\cdots,0}_{p+1}, u_{p+1},\cdots,u_{r-p-1},\underbrace{1,\cdots,1}_{p+1}\}$ and $V = \{\underbrace{0,\cdots,0}_{q+1}, v_{q+1},\cdots,v_{s-q-1},\underbrace{1,\cdots,1}_{q+1}\}$.

Since the boundary knots have multiplicity $p+1$ and $q+1$ with respect to u and v respectively, the surface interpolates the four corner control points. This also implies that the boundary curves depend only on the boundary control points.

3.4.2 B-spline Surface Wavelet Transforms

There are two ways to transform B-spline surfaces. Each transformation is a two-dimensional generalization of the one-dimensional wavelet transform.

The first transform is called the standard decomposition. To obtain the standard decomposition of a B-spline surface, the wavelet transform is first applied to each row of control points. Next, the set of transformed rows is treated as a B-spline surface, and the wavelet transform is applied to each column of control points. The result is a simple-version of the B-spline surface along with its detail coefficients. The algorithm for the standard decomposition is given below.

```
StandardDecomposition:
  $j = level$
  for $row = 0$ to $2^j + d - 1$ do
    Decomposition(ControlPoint)
  end for
  for $column = 0$ to $d$ do
    Decomposition(ControlPoint)
  end for
```

The standard reconstruction is the reverse process of the standard decomposition. The algorithm is given as:

```
StandardReconstruction:
  $j = level$
  for $column = 0$ to $d$ do
    Reconstruction(ControlPoint)
  end for
  for $row = 0$ to $2^j + d - 1$ do
    Reconstruction(ControlPoint)
  end for
```

The second two-dimensional wavelet transform is referred to as the nonstandard decomposition. This transformation alternates the transform operation between rows and columns. First, one step of the wavelet transform is executed in each row of the control points. Next, one step of the wavelet transform is applied to each column of the result. This process is repeated recursively until a simple-version B-spline surface is created. The corresponding algorithm is given below.

```
NonstandardDecomposition:
  $j = level$
  while $j > 0$ do
    for $row = 0$ to $2^j + d - 1$ do
      DecompositionStep(ControlPoint)
    end for
    $j = j - 1$
    for $column = 0$ to $2^j + d - 1$ do
```

```
      DecompositionStep(ControlPoint)
    end for
  end while
```

The pseudo code to perform the nonstandard reconstruction is given below.

```
NonstandardReconstruction:
  $j = 0$
  while $j < level$ do
    for $column = 0$ to $2^j + d - 1$ do
      ReconstructionStep(ControlPoint)
    end for
    $j = j + 1$
    for $row = 0$ to $2^j + d - 1$ do
      ReconstructionStep(ControlPoint)
    end for
  end while
```

For the tensor product surface analysis [37], the synthesis process is given by

$$C^{u+1,v+1} = \left[P^{v+1} Q^{v+1} \right] \begin{bmatrix} C^{u,v} & F^{u,v} \\ G^{u,v} & H^{u,v} \end{bmatrix} \begin{bmatrix} {P^{u+1}}^T \\ {Q^{u+1}}^T \end{bmatrix} \tag{3.11}$$

where u and v are the resolution levels of the control net in the row-wise direction and column-wise direction respectively and $F^{u,v}$, $G^{u,v}$ and $H^{u,v}$ are wavelet coefficient matrices.

The decomposition can be processed with some flexibility. A low-resolution version along the column-wise direction can be obtained from:

$$(P^{v+1^T} P^{v+1})C^{u+1,v} = P^{v+1^T} C^{u+1,v+1} \tag{3.12}$$

The corresponding wavelet coefficients are computed from:

$$(Q^{v+1^T} Q^{v+1})D^{u+1,v} = Q^{v+1^T} (C^{u+1,v+1} - P^{v+1} C^{u+1,v}) \tag{3.13}$$

Similarly, a low-resolution version along the row-wise direction can be found from:

$$C^{u,v+1}(P^{u+1^T} P^{u+1}) = C^{u+1,v+1} P^{u+1} \tag{3.14}$$

The corresponding wavelet coefficients are generated from:

$$D^{u,v+1}(Q^{u+1^T} Q^{u+1}) = (C^{u+1,v+1} - C^{u,v+1} P^{u+1})Q^{u+1} \tag{3.15}$$

The fully low-resolution version is obtained by finding the column-wise coarse version of the row-wise coarse version or the row-wise coarse version of the column-wise coarse version. Equation 3.16 shows how to compute the fully low-resolution version in the matrix representation.

$$(P^{v+1^T} P^{v+1})C^{u,v}(P^{u+1^T} P^{u+1}) = P^{v+1^T} C^{u+1,v+1} P^{u+1} \tag{3.16}$$

CHAPTER 4. MULTIRESOLUTION EDITING FOR B-SPLINE SPACE CURVES AND SURFACES

B-spline wavelets permit multiresolution editing for B-spline curves and surfaces [38]. This can help balance the desire to easily control overall sweep with the need to define localized detail for intuitive B-spline curve and surface modeling. This chapter describes how to apply multiresolution analysis to build B-spline creation and editing techniques that effectively balance the need for local control with the need to control the overall sweep of B-splines.

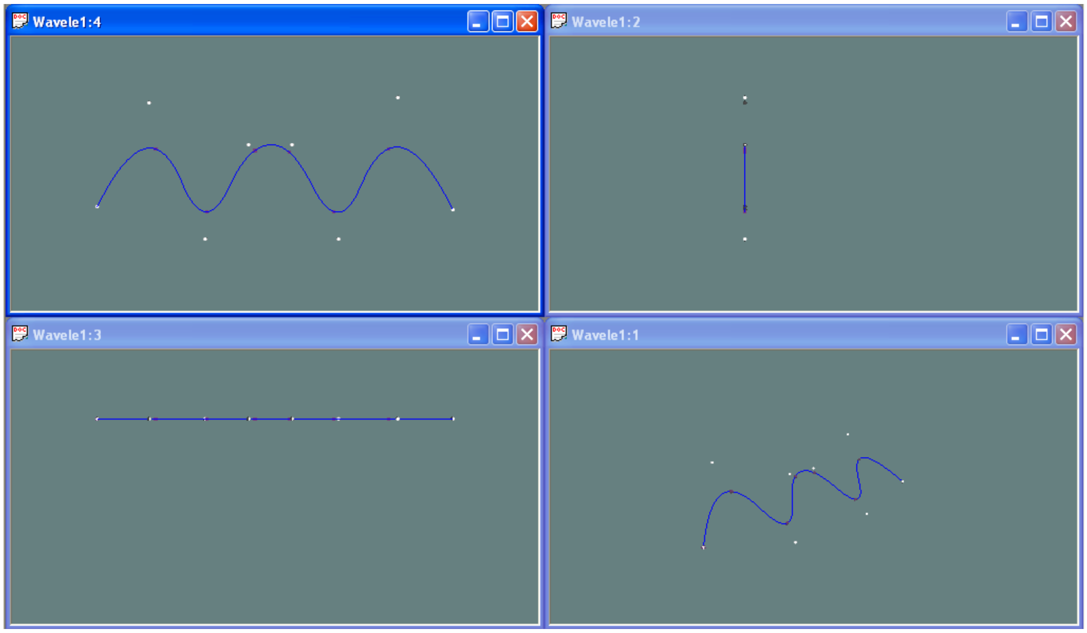
4.1 Conversion from Non-uniform B-splines to Uniform B-splines

The starting point of multiresolution analysis is a nested set of vector spaces. In B-spline wavelets, 2^j $(j = 1,2,\cdots)$ equally spaced interior intervals in B-spline knot vectors are chosen to construct the nested spaces, and B-spline basis functions are defined as the scaling functions. These properties determine that the B-spline multiresolution framework can only be applied for the uniform B-splines with 2^j equally spaced interior intervals. Since non-uniform B-splines are widely used to represent free-form curves and surfaces, this limits the usage of the framework in practice. To overcome this limitation, the research establishes the connection between non-uniform B-splines and the B-spline wavelets, and develops a method to convert non-uniform B-splines to uniform B-splines in the B-spline nested spaces.

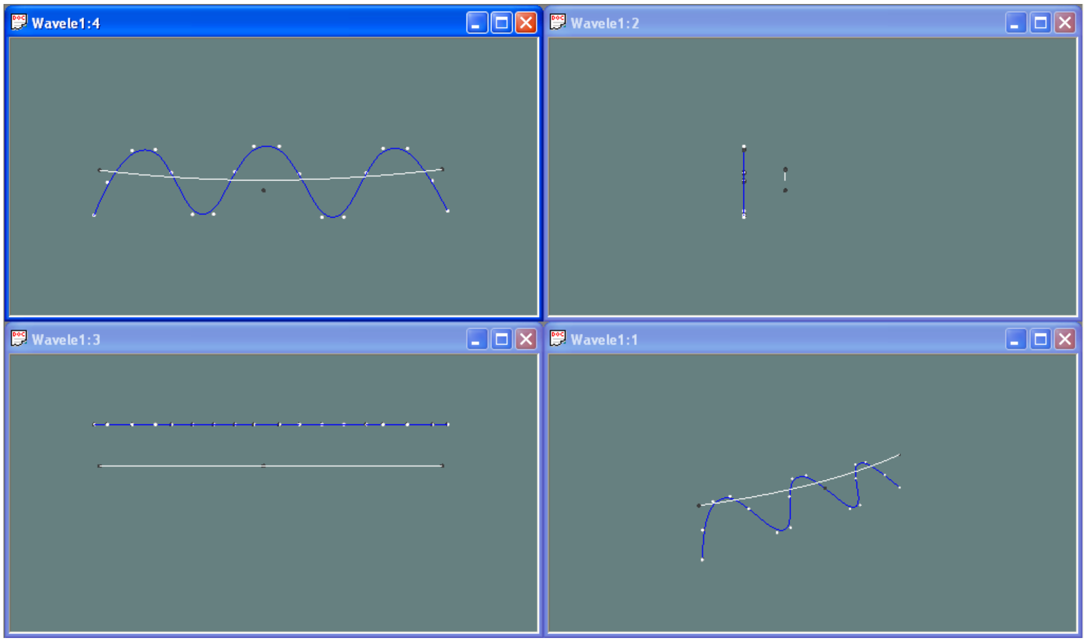
The theory of knot insertion and knot removal for B-splines [2] is employed to approximate non-uniform B-spines in the B-spline nested spaces. Since the vector spaces are defined on the unit interval, a vector space within the spaces can be found to be a good approximation for a non-uniform B-spline representation. The complete set of steps to perform this conversion is shown as follows:

- Find a vector space in the nested vector spaces to approximate a non-uniform B-spline representation. A good approximation is one wherein the distance between any interior knot of the non-uniform B-spline representation and the closest knot of the vector space is within a meaningful tolerance.
- Insert the interior knots in the vector space into the knot vector of the non-uniform B-splines. This upgrades the non-uniform B-spline representation into a higher function space. It is important to note that the B-spline representation is not changed geometrically and parametrically.
- Remove all the interior knots of the original non-uniform B-splines from the upgraded B-splines. This converts the non-uniform B-splines to uniform B-splines in the vector space.

Figures 4.1 - 4.2 illustrate the conversion for non-uniform B-splines. Four viewports are depicted in each figure: Front (top left), Left (top right), Top (bottom left), and Perspective (bottom right). Figure 4.1 shows that a non-uniform B-spline curve is created through global interpolation (see Appendix for details), and then is converted to a uniform B-spline curve at resolution level $j = 4$. Figure 4.2 illustrates that a non-uniform B-spline surface is constructed through a set of section curves (see Appendix for details), and then is converted to a uniform B-spline surface at resolution level ($i = 3, j = 2$).

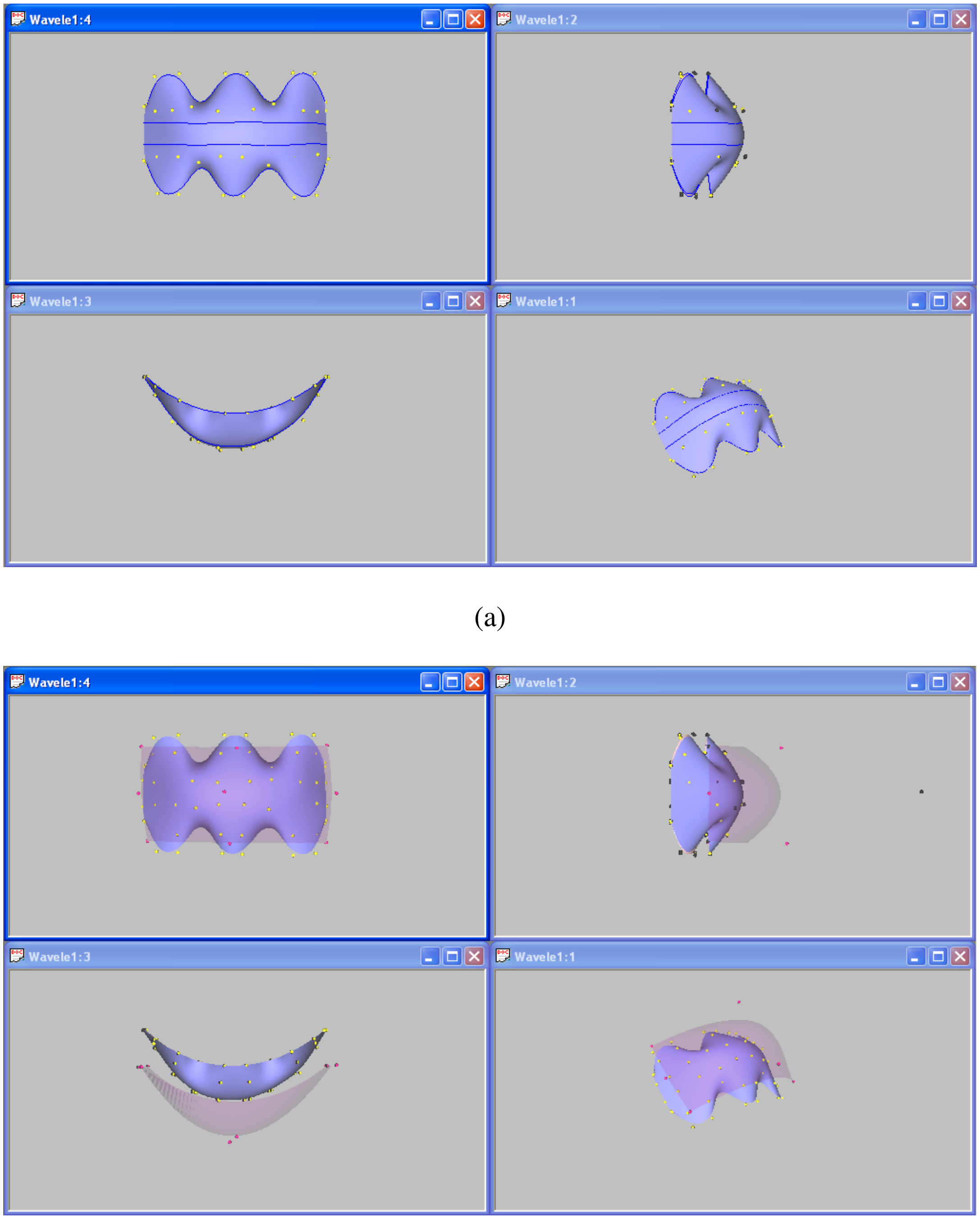


(a)



(b)

Figure 4.1 Conversion from a non-uniform B-spline curve to a uniform B-spline curve in B-spline nested spaces: (a) the blue curve is a non-uniform B-spline curve, created through a set of data points (purple); (b) the non-uniform B-spline curve is converted to a uniform B-spline curve (blue) at resolution level $j = 4$; the white curve is its low resolution version at level $j = 0$.



(a)

(b)

Figure 4.2 Conversion from a non-uniform B-spline surface to a uniform B-spline surface in B-spline nested spaces: (a) the surface is a non-uniform B-spline surface, constructed through a set of section curves (blue); (b) the non-uniform B-spline surface is converted to a uniform B-spline surface (solid) at resolution level ($i = 3$, $j = 2$); the transparent surface is its low resolution version at level ($i = 0$, $j = 0$).

4.2 Multiresolution Editing for B-spline Space Curves

The foregoing chapters describe the recursive process of splitting control points C^j of a B-spline space curve into a lower resolution C^{j-1} and a detail part D^{j-1}. Thus, the original control points C^j can be expressed as a series of lower resolutions $C^0,...,C^{j-1}$ and detail parts $D^0,...,D^{j-1}$. This process is called wavelet decomposition, and is shown schematically in Figure 4.3.

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

Figure 4.3 Decomposition for a B-spline space curve.

The original control points C^j can be recovered from the sequence $C^0, D^0, D^1 ..., D^{j-1}$, which is known as a wavelet transform. The process is called wavelet reconstruction, and is shown schematically in Figure 4.4.

$$\begin{tikzcd}
C^j & C^{j-1} \arrow[l, "P^j"'] & C^{j-2} \arrow[l, "P^{j-1}"'] & \cdots & C^1 & C^0 \arrow[l, "P^1"'] \\
 & D^{j-1} \arrow[ul, "Q^j"'] & D^{j-2} \arrow[ul, "Q^{j-1}"'] & \cdots & & D^0 \arrow[ul, "Q^1"']
\end{tikzcd}$$

Figure 4.4 Reconstruction for a B-spline space curve.

Wavelet decomposition and reconstruction, two important processes in multiresolution analysis, provide the basis for multiresolution editing of B-spline space curves.

4.2.1 Overall Sweep Editing for B-spline Space Curves

Multiresolution editing allows for changing the overall sweep of a B-spline space curve while maintaining its fine details. In contrast to traditional editing methods where many control points need to be manipulated to deform a B-spline representation with complex details, multiresolution editing methods can achieve the same effect by editing only several control points at a low resolution level.

B-spline function spaces are constructed through a series of uniformly spaced knot sequences. Subspaces are defined by removing the midpoint knot for each adjacent pair of interior intervals, so the knot span of the basis function in a spline space is only half as long as that of the basis function in its immediate subspace. This means that editing a single control point affects a larger portion of the curve in the subspace. At the lowest resolution level, the curve contains no interior knots, and editing a single control point affects the shape of the entire curve. At the highest resolution level, editing a single control point influences only a narrow portion of the curve.

Let C^j be the original control points of a B-spline space curve $f^j(u)$. Assume that C^k is a lower resolution version of C^j, and ΔC^k is an edited part at the lower resolution level. The edited curve at the highest resolution can be computed through reconstruction:

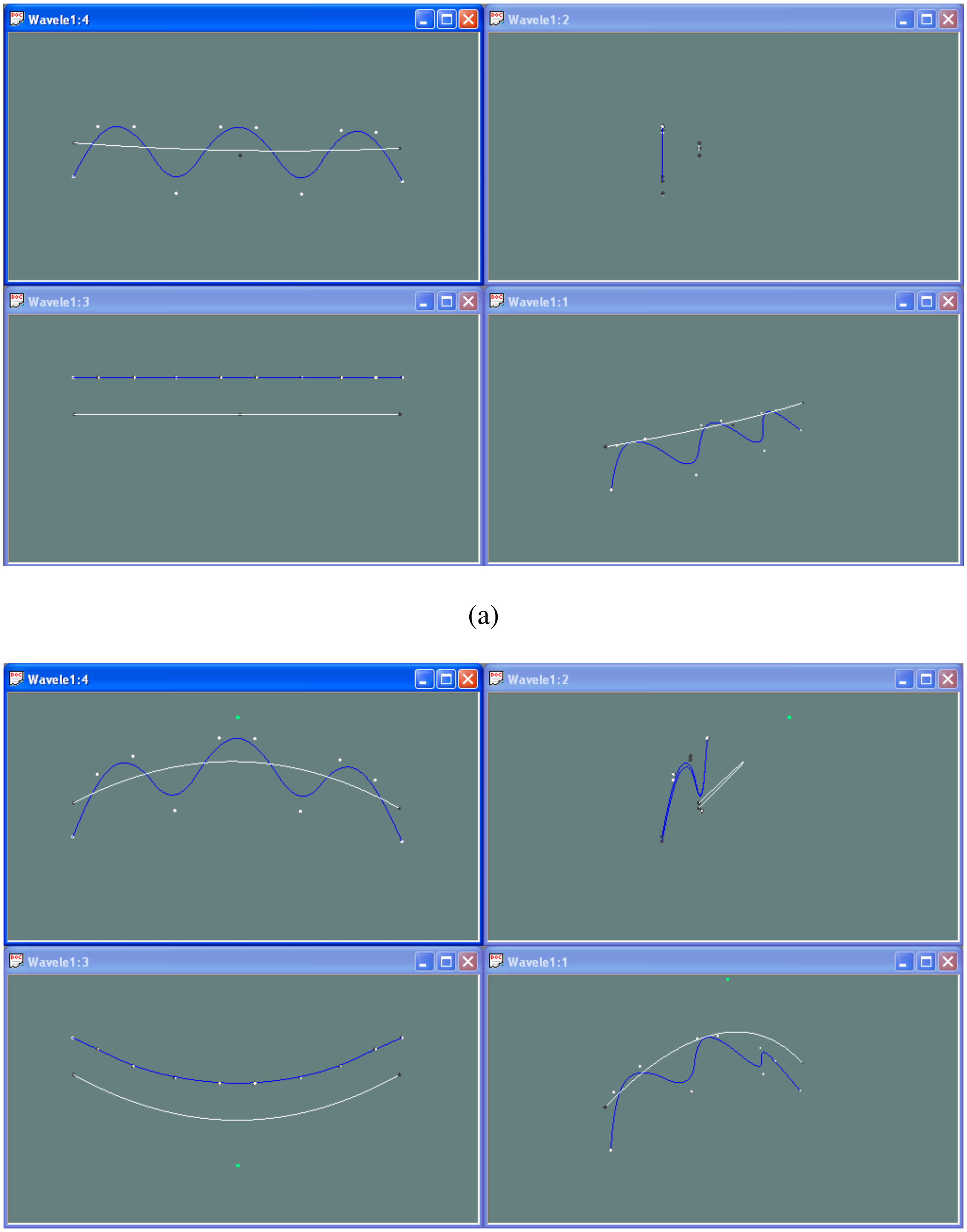
$$\overline{C}^j = C^j + \Delta C^j = C^j + P^j P^{j-1} \ldots P^{k+1} \Delta C^k \qquad (4.1)$$

Control point editing at any particular resolution level invalidates the control points at all other resolution levels. To maintain consistency in the multiresolution representation, the new control points at the edited resolution level must be decomposed to lower resolution levels and reconstructed to higher resolution levels. Figure 4.5 show a schematic example of the process at editing level k .

$$\begin{tikzcd}
 & & & \Delta C^{k} \arrow[d] & & & \\
\overline{C}^{j} & \overline{C}^{j-1} \arrow[l] & \overline{C}^{j-2}\cdots \arrow[l] & C^{k} \arrow[r] \arrow[dr] & \overline{C}^{k-1}\cdots & \overline{C}^{1} \arrow[r] \arrow[dr] & \overline{C}^{0} \\
 & D^{j-1} \arrow[ul] & D^{j-2}\cdots \arrow[ul] & & \overline{D}^{k-1}\cdots & & \overline{D}^{0}
\end{tikzcd}$$

Figure 4.5 Overall sweep editing at level k .

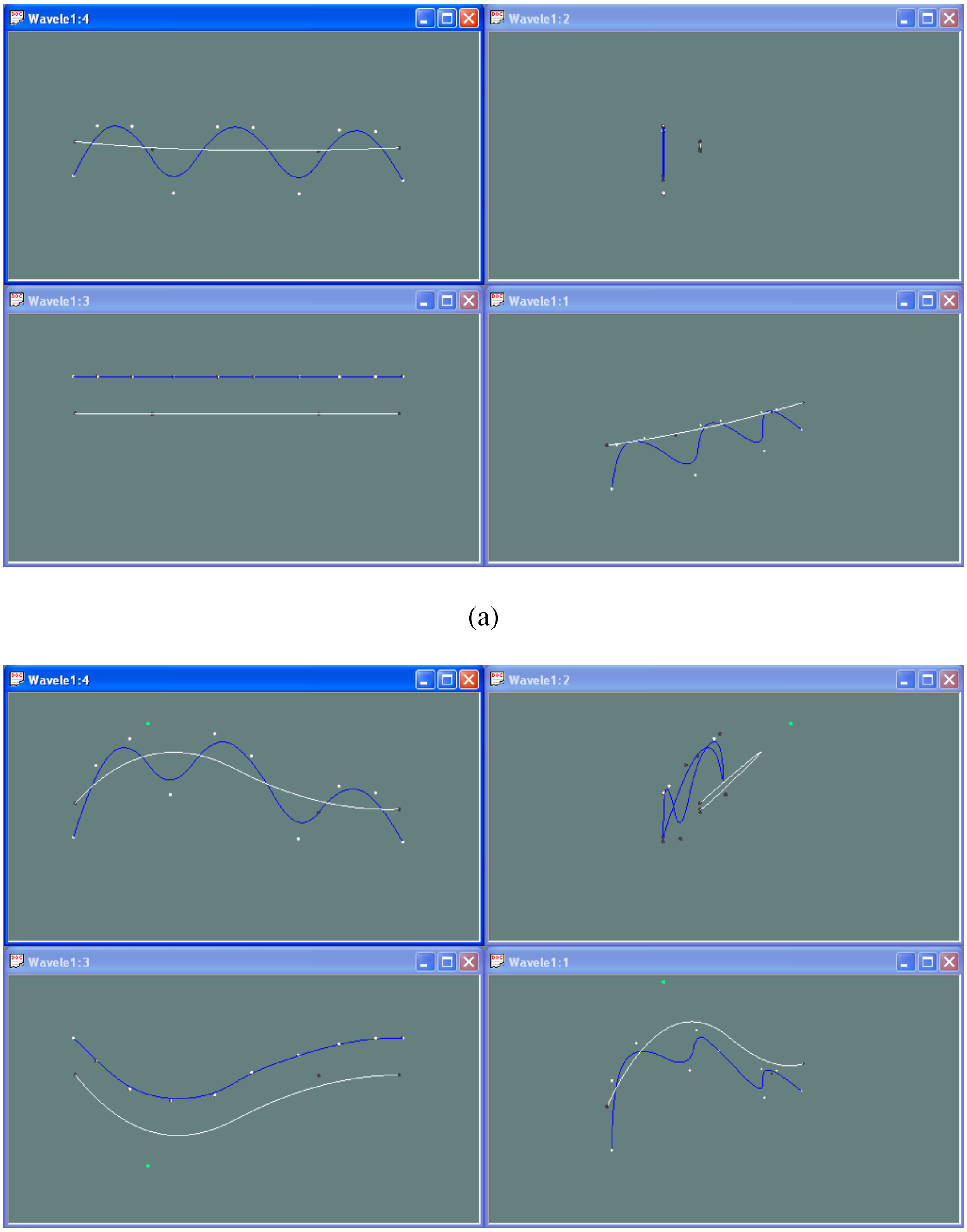
During the editing process, modification of the control points ΔC^{k} changes the overall sweep of the B-spline space curve, and the details $D^{k}\cdots,D^{j-1}$ preserve the original fine details on the curve. The decomposition process splits the edited version $C^{k}+\Delta C^{k}$ into new low resolution versions $\overline{C}^{k-1}\cdots,\overline{C}^{0}$ and new details $\overline{D}^{k-1}\cdots,\overline{D}^{0}$ **.** Figures 4.6 - 4.8 represent an example of multiresolution decomposition for a B-spline space curve, and show the overall sweep editing at different resolution levels. Four viewports are depicted in each figure: Front (top left), Left (top right), Top (bottom left), and Perspective (bottom right). The original curve and its low resolution version are shown in (a). The overall sweep of the curve after editing one control point at lower level is shown in (b).



(a)

(b)

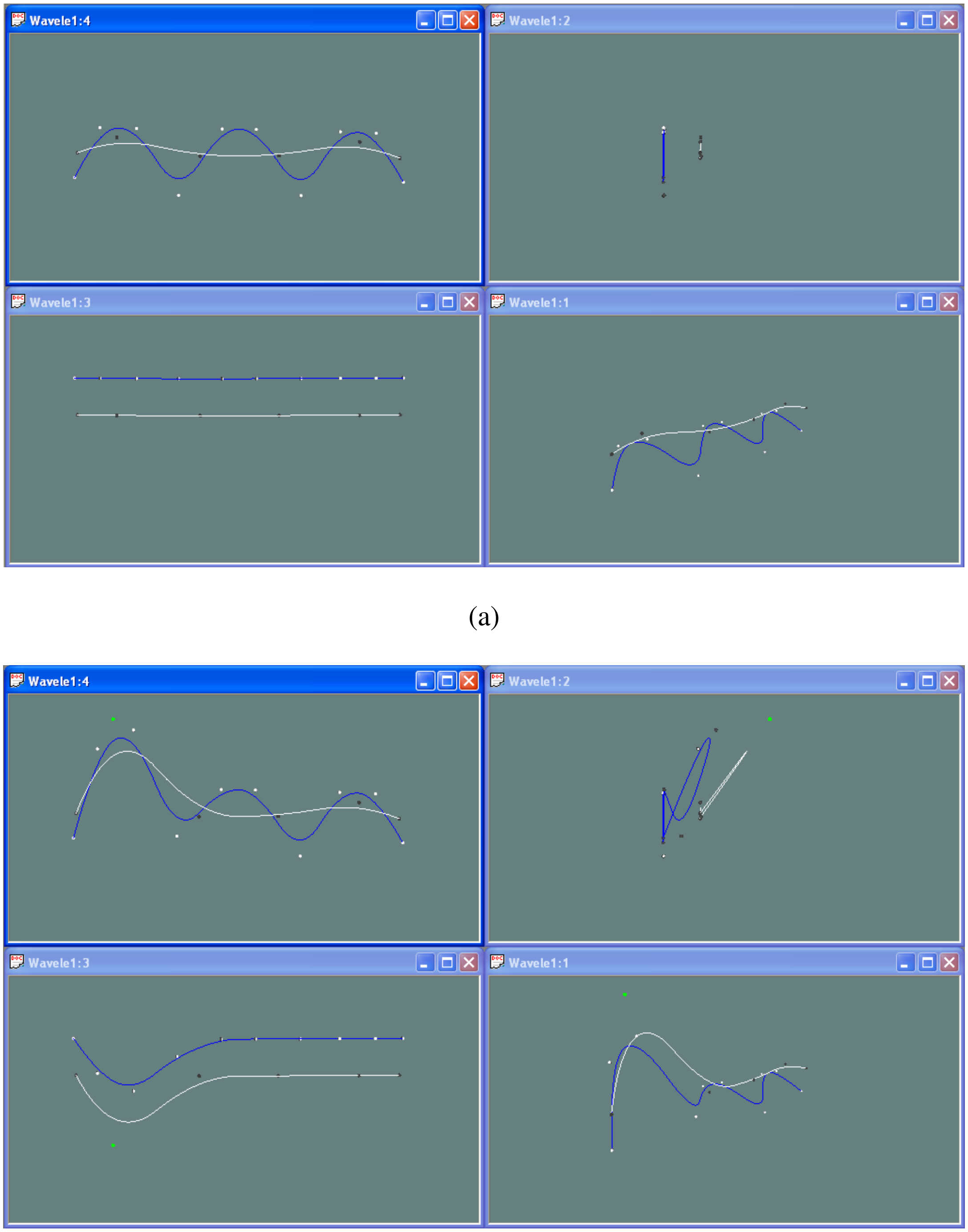
Figure 4.6 Overall sweep editing: there are four resolution levels ($j = 0, 1, 2, 3$) at the original curve; (a) the blue curve is the one at the highest level ($j = 3$), and the white curve is the one at the lowest level ($j = 0$); (b) the overall sweep of the curve after editing one control point (green) at level $j = 0$.



(a)

(b)

Figure 4.7 Overall sweep editing: there are four resolution levels ($j = 0,1,2,3$) at the original curve; (a) the blue curve is the one at the highest level ($j = 3$), and the white curve is the one at lower level ($j = 1$); (b) the overall sweep of the curve after editing one control point (green) at level $j = 1$.



(a)

(b)

Figure 4.8 Overall sweep editing: there are four resolution levels ($j = 0,1,2,3$) at the original curve; (a) the blue curve is the one at the highest level ($j = 3$), and the white curve is the one at lower level ($j = 2$); (b) the overall sweep of the curve after editing one control point (green) at level $j = 2$.

4.2.2 Detail Editing for B-spline Space Curves

Another form of editing that is naturally supported by multiresolution analysis is one of editing the detail parts of a B-spline space curve without affecting its overall sweep. Finkelstein and Salesin introduced a method to edit the character of a curve by replacing the existing set of details $D^0, ..., D^{j-1}$ with some new set $\overline{D}^0, ..., \overline{D}^{j-1}$. As previously discussed, this approach does not produce an interactive and intuitive editing interface. Moreover, the wavelet representations of detail tend to behave in undesirable ways during editing [14].

To address these deficiencies, this research introduces the knot refinement method of B-splines into multiresolution analysis, and establishes the connection between multiresolution analysis and B-spline subdivision to define different levels of resolution. This method facilitates control point editing and provides an intuitive and convenient interface for interactive detail editing.

4.2.2.1 Knot Refinement

It is often necessary to simultaneously insert several knots into the knot vector of a B-spline; the method is known as *knot refinement* [2]. Let $f(u) = \sum_{i=0}^{n} N_{i,d}(u) \cdot c_i$ be a B-spline curve defined on the knot vector $U = \{u_0, \cdots, u_m\}$, and let $t = \{t_0, \cdots, t_r\}$ be a series of non-decreasing values in (u_0, u_m). The elements of t are to be inserted into the knot vector $U = \{u_0, \cdots, u_m\}$ to form a new knot vector $\overline{U}$. It is obvious that the vector space defined on the original knot vector U is a subspace of the vector space defined on the new knot vector $\overline{U}$, so the curve $f(u)$ has a representation in the space $V_{\overline{u}}$

$$f(u) = \sum_{0}^{n+r+1} \overline{N}_{i,d}(u) \cdot \overline{c_i} \qquad (4.2)$$

where the $\{\overline{N}_{i,d}(u)\}$ are the basis functions on the knot vector $\overline{U}$.

The process of computing the new control points $\{\overline{c_i}\}$ in Equation 4.2 is referred to as knot refinement. It is important to note that knot refinement extends the curve into a higher resolution space. The space curve itself is not changed either geometrically or parametrically. A classic knot refinement algorithm is given by Boehm and Prautzsch [39].

4.2.2.2 B-spline Wavelets and Subdivision

From the B-spline local support property, it is clear that the knot span for a basis function $N_{i,d}(u)$ is within the interval $[u_i, u_{i+d+1})$. This means that it is impossible to edit fine details within the interval $[u_i, u_{i+d+1})$ since each control point affects the entire span. This results in limitations in detail editing for a B-spline curve. To overcome this limitation, the knot refinement is applied to introduce sufficient resolution in the curve to facilitate editing at any desired level of detail.

Let $f^j(u) = \sum_{i=0}^{2^j+d-1} N_{i,d}^j(u) \cdot c_i^j$ be a B-spline curve defined at resolution level j, with the knot vector $U = 1/2^j \{\underbrace{0, \cdots 0,}_{d+1}\ 1,\ 2,\ \cdots,\ 2^j - 1,\ \underbrace{2^j,\ \cdots,\ 2^j}_{d+1}\}$. For each interior interval $[u_i, u_{i+1})$, the midpoint of the interval is used as a new inserted knot. Using the knot refinement method, the curve is extended to resolution level $j+1$, and editing one control point at this level will influence a curve region that is only half the size of the region influenced by the editing of a single control point at resolution level j. The curve is not changed geometrically and parametrically, so any former editing information is preserved. In matrix notation, this can be expressed as

$$N^{j}(u)\cdot C^{j} = N^{j+1}(u)\cdot C^{j+1} \tag{4.3}$$

This process can be repeated recursively, and it creates a subdivision scheme. Recalling nested spaces in developing B-spline wavelets, indicates that the subdivision scheme leads to such a sequence of spaces. This means that the basis functions in the subdivision scheme are refinable; that is, each basis function at resolution level j can be expressed as a linear combination of the basis functions at resolution level $j+1$. Referring to Equation 3.3, this can be written as

$$N^{j}(u) = N^{j+1}(u)P^{j+1} \tag{4.4}$$

Substituting Equation 4.4 into Equation 4.3 gives

$$C^{j+1} = P^{j+1}C^{j} \tag{4.5}$$

As a result, the constant matrix P^{j} serves as the synthesis matrix for the scaling functions $\Phi^{j}(x)$ and the subdivision matrix for the control points C^{j}.

The subdivision can be applied to the curve recursively until detail editing on the curve meets the user's requirements. This process refines the curve to a higher function space where the required fine detail features can be achieved. Since the subdivision scheme leads to the B-spline nested spaces, B-spline wavelet decomposition can be applied to the refined curve. The detail features are edited at the highest resolution level, and the overall sweep is changed by multiresolution editing at low resolution levels. Therefore, the detail editing method effectively manages the balance between local and global control for B-splines.

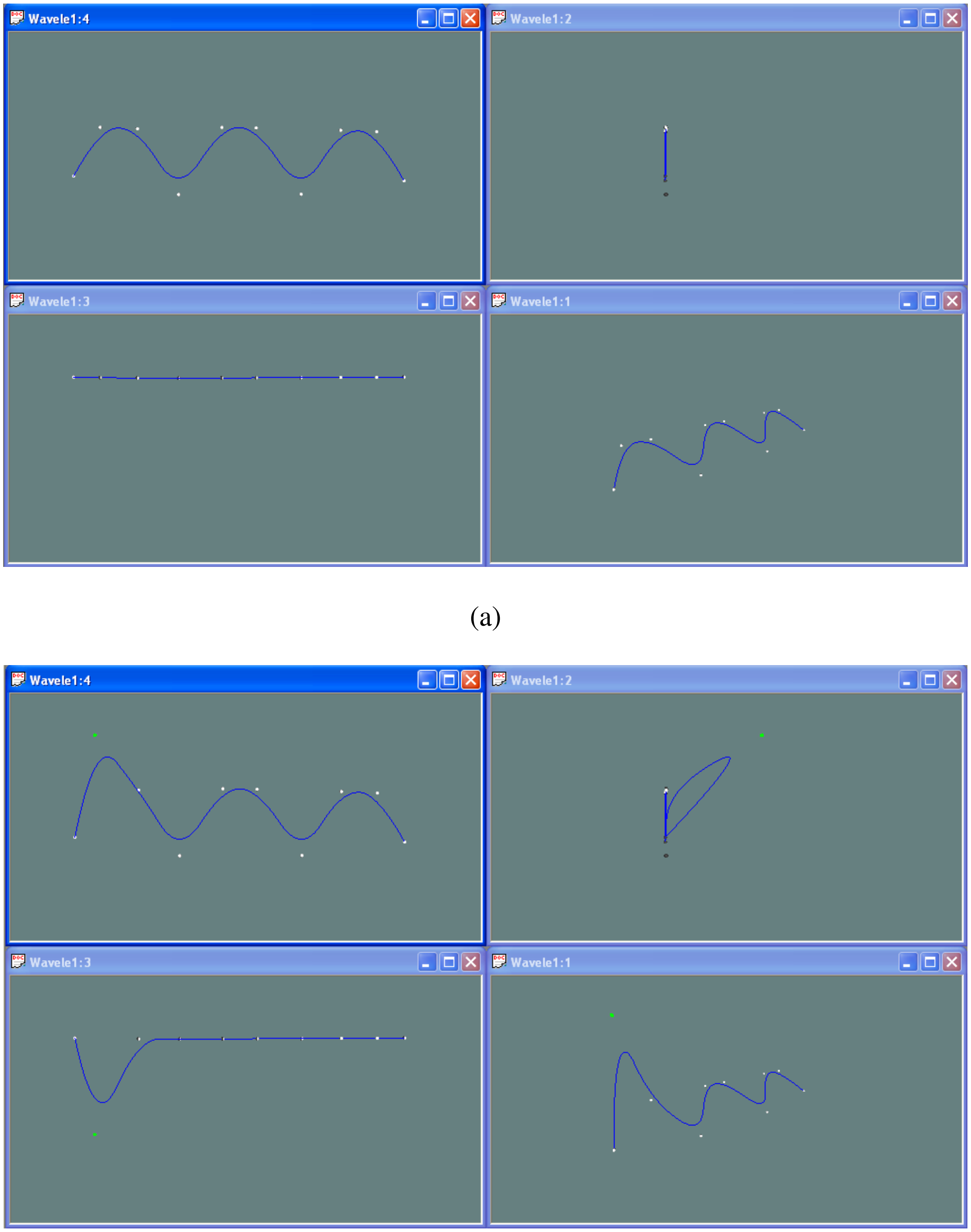
Figure 4.9 shows the subdivision procedure for extending a B-spline curve to a high resolution level. Figure 4.10 shows detail editing at the extended high resolution level. Figures 4.11 - 4.13 illustrate detail editing for a B-spline space curve at different resolution levels. Four viewports are depicted in each figure: Front (top left), Left (top right), Top (bottom left), and Perspective (bottom right). The subdivision curve is shown in (a), and detail editing at the subdivision curve is shown in (b). Figure 4.14 shows an example of global control for the subdivision curve.

$$\begin{tikzcd}
& \textit{subdivision} \arrow[d] & & & & \textit{subdivision} \arrow[d] \\
& P^n & & & & P^{j+1} \\
C^n & & C^{n-1} \arrow[ll] & \cdots & C^{j+1} & & C^j \arrow[ll] \arrow[r] \arrow[rd] & C^{j-1} & \cdots & C^1 \arrow[r] \arrow[rd] & C^0 \\
& & & & & & & D^{j-1} & \cdots & & D^0
\end{tikzcd}$$

Figure 4.9 Subdivision procedure to extend a curve to the high resolution level n.

$$\begin{tikzcd}
\Delta C^n \arrow[d] \\
C^n \arrow[r] \arrow[rd] & \overline{C}^{n-1} & \cdots & \overline{C}^{j+1} \arrow[r] \arrow[rd] & \overline{C}^{j} \arrow[r] \arrow[rd] & \overline{C}^{j-1} & \cdots & \overline{C}^{1} \arrow[r] \arrow[rd] & \overline{C}^{0} \\
& \overline{D}^{n-1} & & & \overline{D}^{j} & \overline{D}^{j-1} & \cdots & & \overline{D}^{0}
\end{tikzcd}$$

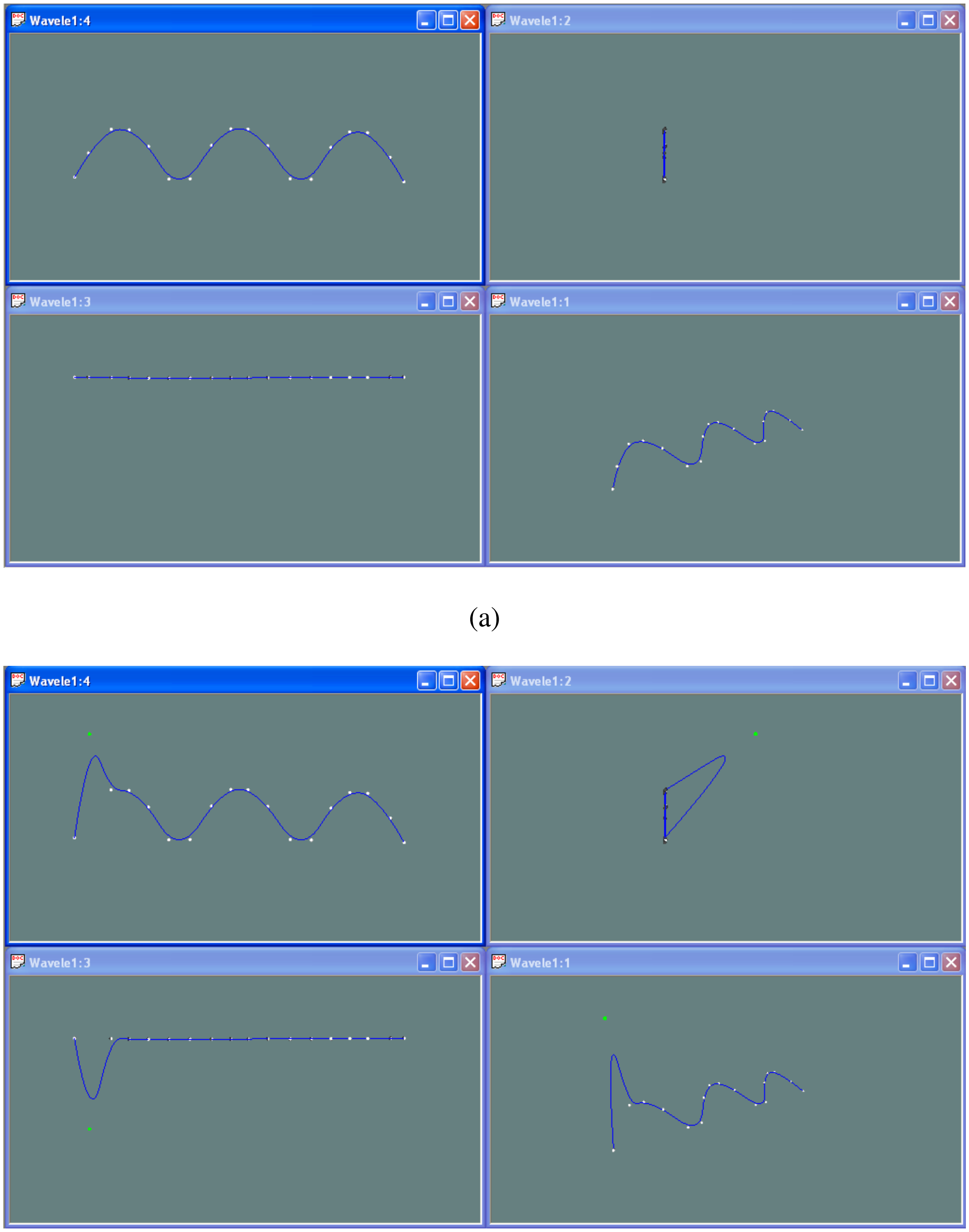
Figure 4.10 Detail editing at the extended resolution level n.



(a)

(b)

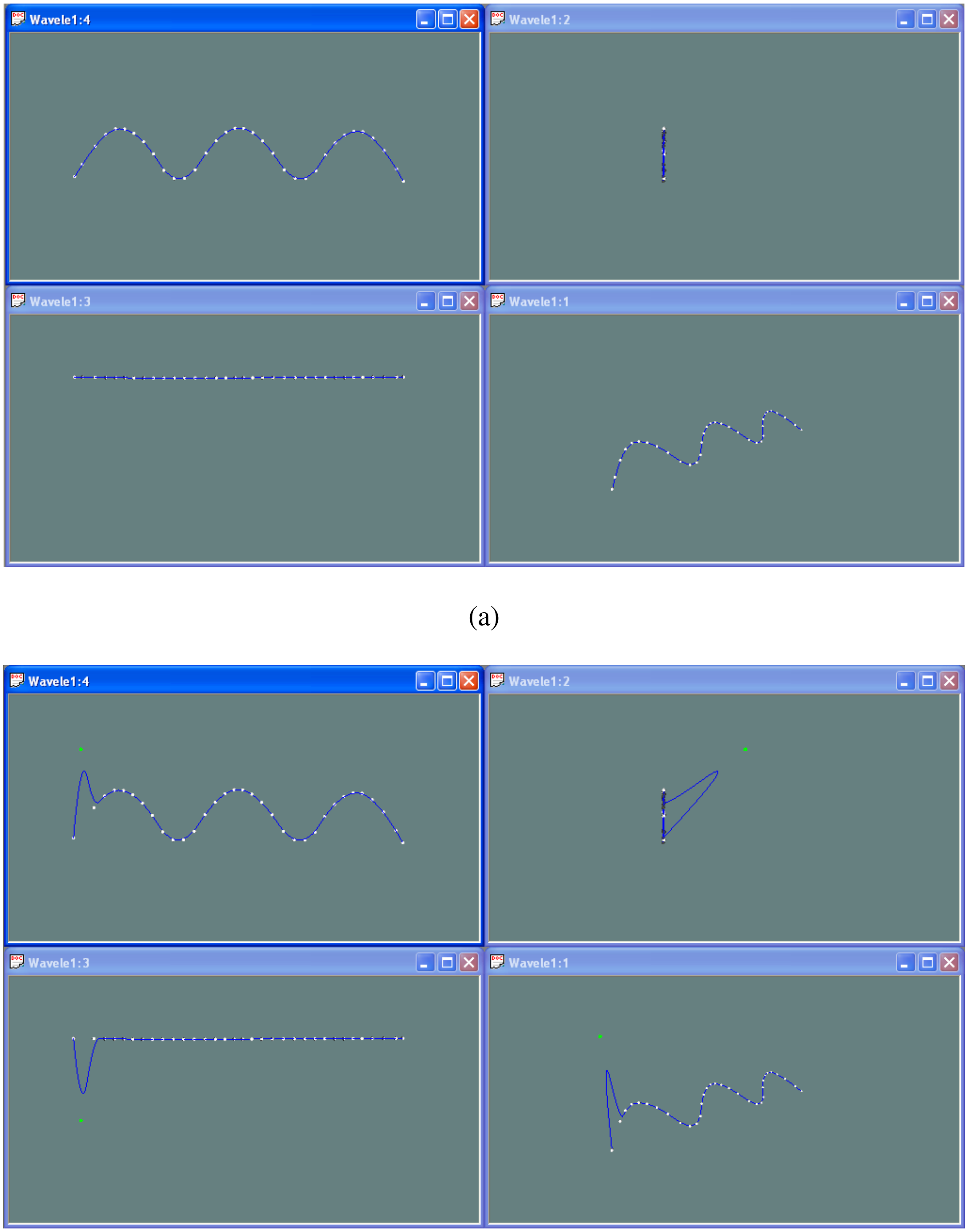
Figure 4.11 Detail editing: there are four resolution levels ($j = 0,1,2,3$) at the original curve; (a) the blue curve is the one at the highest level ($j = 3$); (b) detail editing after editing one control point (green) at level $j = 3$.



(a)

(b)

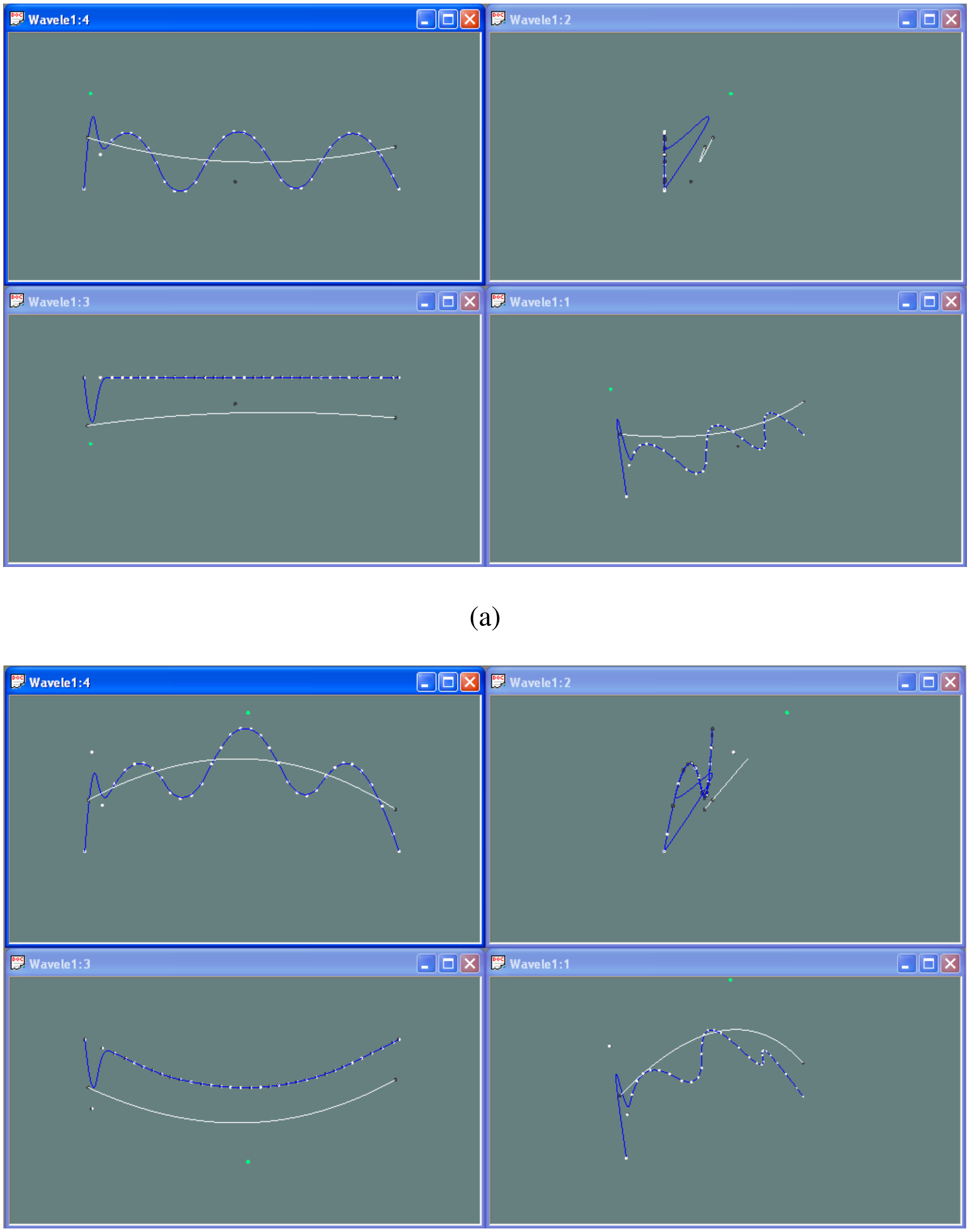
Figure 4.12 Detail editing: there are four resolution levels ($j = 0, 1, 2, 3$) at the original curve; (a) the blue curve is the curve after one step subdivision; the curve is refined to next higher resolution level ($j = 4$); (b) detail editing after editing one control point (green) at level $j = 4$.



(a)

(b)

Figure 4.13 Detail editing: there are four resolution levels ($j = 0, 1, 2, 3$) at the original curve; (a) the blue curve is the curve after two step subdivisions; the curve is refined to next two higher resolution level ($j = 5$); (b) detail editing after editing one control point (green) at level $j = 5$.



(a)

(b)

Figure 4.14 Global control for the subdivision curve: there are four resolution levels ($j = 0, 1, 2, 3$) at the original curve; the blue curve is the curve after two step subdivisions; (a) detail editing after editing one control point (green) at level $j = 5$; (b) overall sweep editing after editing one control point (green) at level $j = 0$.

4.3 Multiresolution Editing for B-spline Surfaces

As discussed in section 3.4, the one-dimensional multiresolution scheme can be applied to 3D tensor product surfaces. A tensor product B-spline surface is defined on a bidirectional control net. This allows the surface to be denoted in the u and v directions. Each row aligned along the u direction is referred to as a v-curve, and each column aligned along the v direction is a u-curve. To apply multiresolution analysis to the surface, all u and v curves can be considered as independent curves to which the analysis and synthesis algorithms may be applied. For example, to decompose a B-spline surface, the analysis filter is applied on all rows in the control net, and then on all columns in the smaller control net that results from reducing the resolution of all rows. Figure 4.15 shows the decomposition process for a tensor product surface. To allow the user to edit at any resolution level, all kinds of reconstruction situations are included, as shown in Figure 4.16.

$$\begin{tikzcd}
 & D^{u-1,v} & & D^{u-2,v} & & & & & D^{0,v} & & \\
C^{u,v} \arrow[ur] \arrow[rr] \arrow[dr] \arrow[dd] & & C^{u-1,v} \arrow[ur] \arrow[rr] \arrow[dr] \arrow[dd] & & C^{u-2,v} \arrow[dr] \arrow[dd] & & \cdots & C^{1,v} \arrow[ur] \arrow[rr] \arrow[dr] \arrow[dd] & & C^{0,v} \arrow[dr] \arrow[dd] & \\
 & D^{u,v-1} & & D^{u-1,v-1} & & D^{u-2,v-1} & & & D^{1,v-1} & & D^{0,v-1} \\
C^{u,v-1} & & C^{u-1,v-1} & & C^{u-2,v-1} & & & C^{1,v-1} & & C^{0,v-1} & \\
\vdots & & \vdots & & \vdots & & & \vdots & & \vdots & \\
C^{u,1} \arrow[dr] \arrow[dd] & & C^{u-1,1} \arrow[dr] \arrow[dd] & & C^{u-2,1} \arrow[dr] \arrow[dd] & & & C^{1,1} \arrow[dr] \arrow[dd] & & C^{0,1} \arrow[dr] \arrow[dd] & \\
 & D^{u,0} & & D^{u-1,0} & & D^{u-2,0} & & & D^{1,0} & & D^{0,0} \\
C^{u,0} & & C^{u-1,0} & & C^{u-2,0} & & & C^{1,0} & & C^{0,0} &
\end{tikzcd}$$

Figure 4.15 Decomposition for a tensor product B-spline surface.

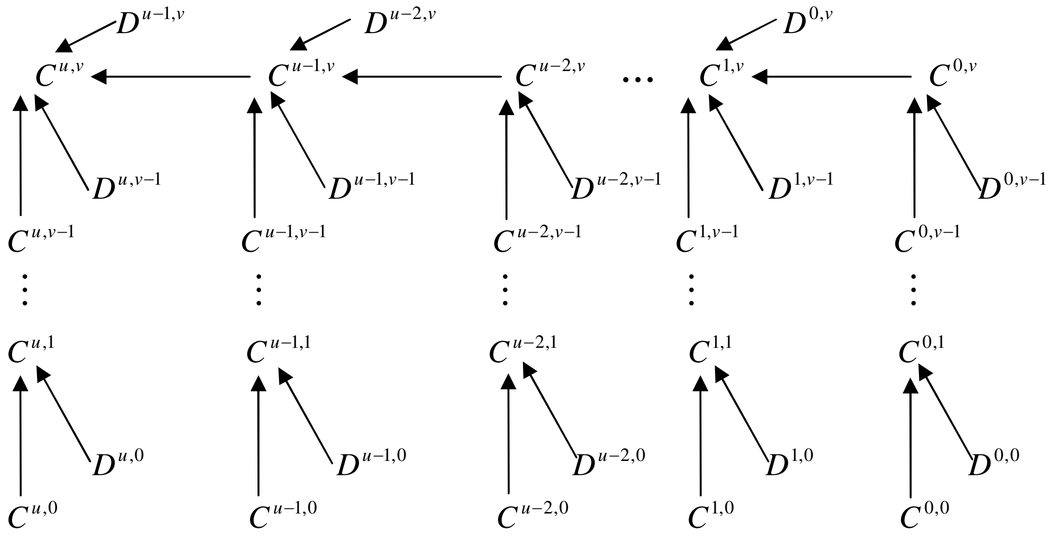


Figure 4.16 Reconstruction for a tensor product B-spline surface.

4.3.1 Overall Sweep Editing for B-spline Surfaces

The method for editing the overall sweep of a B-spline surface is a straightforward extension of the curve-editing method. Since editing on a B-spline surface can be executed in the u and v directions, this gives the user greater freedom with respect to controlling the overall sweep of the surface. For instance, if the user chooses the lowest resolution level in the u direction, any control point editing at this level will affect the entire shape of the surface along the u direction. The user can choose different resolution levels in the v direction to independently determine the affected portion of the surface in the v direction. At the lowest level in the v direction, the entire shape in that direction is affected. At the highest level in the v direction, only the local narrow portion in that direction is changed.

The complete set of steps needed to perform overall sweep editing is shown as follows:

Step 1: Control point editing at a resolution level. Figure 4.17 shows an example of overall sweep editing for a surface at resolution level $(u-2,0)$.

$$\begin{tikzcd}
 & D^{u-1,v} & & D^{u-2,v} & & & & & D^{0,v} & & \\
C^{u,v} \arrow[ur] \arrow[rr] \arrow[dd] \arrow[dr] & & C^{u-1,v} \arrow[ur] \arrow[rr] \arrow[dd] \arrow[dr] & & C^{u-2,v} \arrow[dd] \arrow[dr] & & \cdots & C^{1,v} \arrow[ur] \arrow[rr] \arrow[dd] \arrow[dr] & & C^{0,v} \arrow[dd] \arrow[dr] & \\
 & D^{u,v-1} & & D^{u-1,v-1} & & D^{u-2,v-1} & & & D^{1,v-1} & & D^{0,v-1} \\
C^{u,v-1} & & C^{u-1,v-1} & & C^{u-2,v-1} & & & C^{1,v-1} & & C^{0,v-1} & \\
\vdots & & \vdots & & \vdots & & & \vdots & & \vdots & \\
C^{u,1} \arrow[dd] \arrow[dr] & & C^{u-1,1} \arrow[dd] \arrow[dr] & & C^{u-2,1} \arrow[dd] \arrow[dr] & & & C^{1,1} \arrow[dd] \arrow[dr] & & C^{0,1} \arrow[dd] \arrow[dr] & \\
 & D^{u,0} & & D^{u-1,0} & & D^{u-2,0} & & & D^{1,0} & & D^{0,0} \\
C^{u,0} & & C^{u-1,0} & & C^{u-2,0} & & & C^{1,0} & & C^{0,0} & \\
 & & & & & \Delta C^{u-2,0} \arrow[ul, red] & & & & &
\end{tikzcd}$$

Figure 4.17 Overall sweep editing for a surface at resolution level $(u-2, 0)$.

Step 2: Reconstruction from the edited resolution level. The original details along the construction path are used to preserve the original fine details on the surface.

$$\begin{tikzcd}
 & D^{u-1,v} \arrow[dl] & & D^{u-2,v} \arrow[dl] & & \\
\overline{C}^{u,v} & & \overline{C}^{u-1,v} \arrow[ll] & & \overline{C}^{u-2,v} \arrow[ll] & \\
 & & & & & D^{u-2,v-1} \arrow[ul] \\
 & & & & \overline{C}^{u-2,v-1} \arrow[uu] & \\
 & & & & \vdots & \\
 & & & & \overline{C}^{u-2,1} & \\
 & & & & & D^{u-2,0} \arrow[ul] \\
 & & & & C^{u-2,0} + \Delta C^{u-2,0} \arrow[uu] &
\end{tikzcd}$$

Figure 4.18 Reconstruction from the edited level $(u-2, 0)$ to the highest level (u,v).

Step 3: Decomposition for the surface. This process updates all the resolution levels except those on the construction path in step 2, as shown in Figure 4.19.

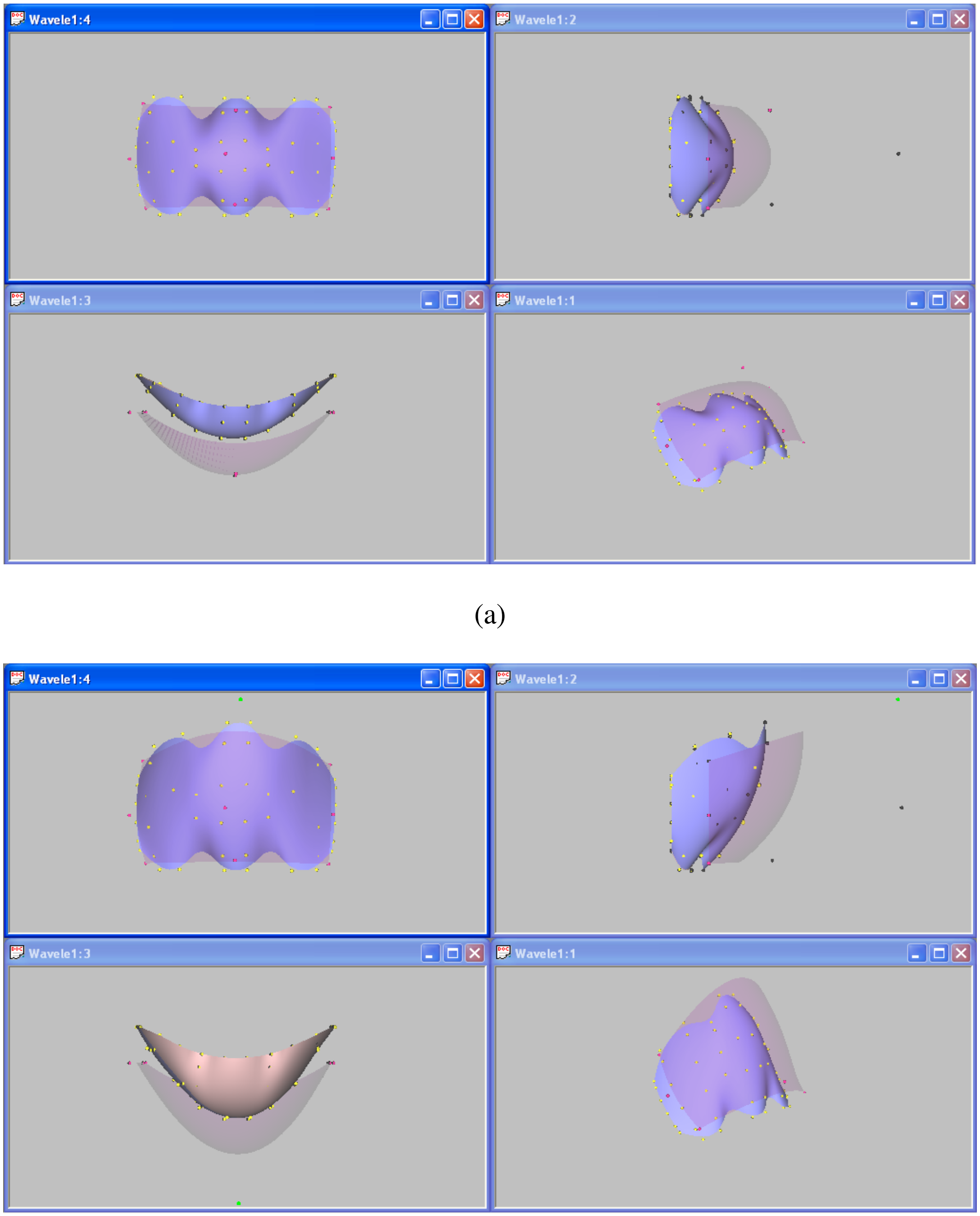
$$\begin{tikzcd}
& D^{u-1,v} & & D^{u-2,v} & & & & & \overline{D}^{0,v} & & \\
\overline{C}^{u,v} \arrow[ur] \arrow[rr] \arrow[dr] \arrow[dd] & & \overline{C}^{u-1,v} \arrow[ur] \arrow[rr] \arrow[dr] \arrow[dd] & & \overline{C}^{u-2,v} \arrow[dr] \arrow[dd] & & \cdots & \overline{C}^{1,v} \arrow[ur] \arrow[rr] \arrow[dr] \arrow[dd] & & \overline{C}^{0,v} \arrow[dr] \arrow[dd] & \\
& \overline{D}^{u,v-1} & & \overline{D}^{u-1,v-1} & & D^{u-2,v-1} & & & \overline{D}^{1,v-1} & & D^{0,v-1} \\
\overline{C}^{u,v-1} & & \overline{C}^{u-1,v-1} & & \overline{C}^{u-2,v-1} & & & \overline{C}^{1,v-1} & & \overline{C}^{0,v-1} & \\
\vdots & & \vdots & & \vdots & & & \vdots & & \vdots & \\
\overline{C}^{u,1} \arrow[dr] \arrow[dd] & & \overline{C}^{u-1,1} \arrow[dr] \arrow[dd] & & \overline{C}^{u-2,1} \arrow[dr] \arrow[dd] & & & \overline{C}^{1,1} \arrow[dr] \arrow[dd] & & \overline{C}^{0,1} \arrow[dr] \arrow[dd] & \\
& \overline{D}^{u,0} & & \overline{D}^{u-1,0} & & D^{u-2,0} & & & \overline{D}^{1,0} & & \overline{D}^{0,0} \\
\overline{C}^{u,0} & & \overline{C}^{u-1,0} & & \overline{C}^{u-2,0} & & & \overline{C}^{1,0} & & \overline{C}^{0,0} &
\end{tikzcd}$$

Figure 4.19 Decomposition for the surface.

Figures 4.20 - 4.23 represent an example of multiresolution decomposition for a B-spline surface, and show the overall sweep editing at different resolution levels. Four viewports are depicted in each figure: Front (top left), Left (top right), Top (bottom left), and Perspective (bottom right). The original surface and its low resolution version are shown in (a). The overall sweep of the surface after editing one control point at lower level is shown in (b).

4.3.2 Detail Editing for B-spline Surfaces

The one-dimensional detail editing method can be extended to two-dimensional situations. Subdivision can recursively be applied to a tensor product surface in the u or v direction until detail editing on the surface meets fine detail requirements. The user determines the subdivision direction and the steps of the subdivision in that direction. The



(a)

(b)

Figure 4.20 Overall sweep editing: there are four resolution levels ($i = 0, 1, 2, 3$) in the u direction and three resolution levels ($j = 0, 1, 2$) in the v direction at the original surface; (a) the solid surface is the one at the highest level ($i = 3$, $j = 2$); the transparent surface is the one at the lowest level ($i = 0$, $j = 0$); (b) the overall sweep of the surface after editing one control point (green) at level ($i = 0$, $j = 0$).

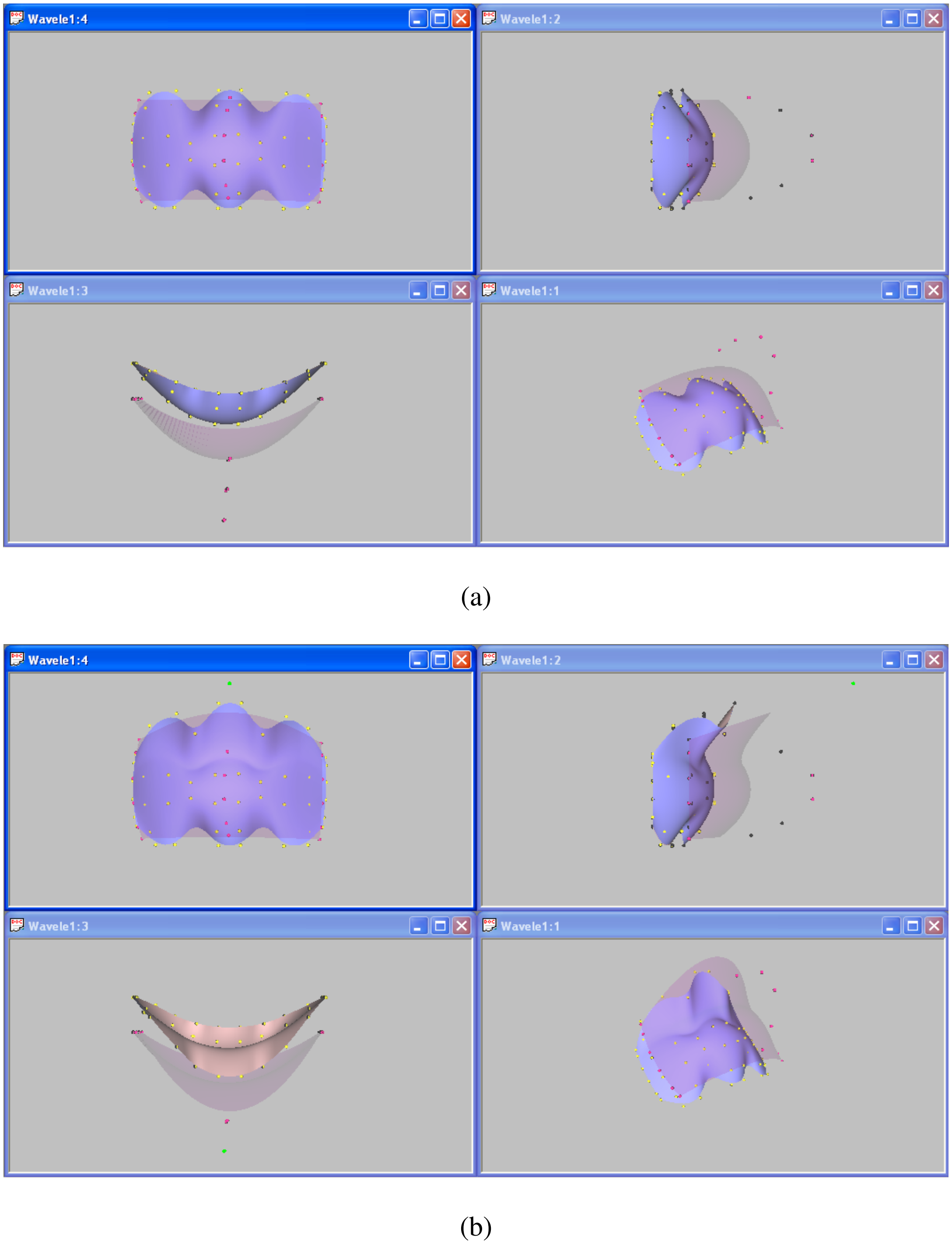
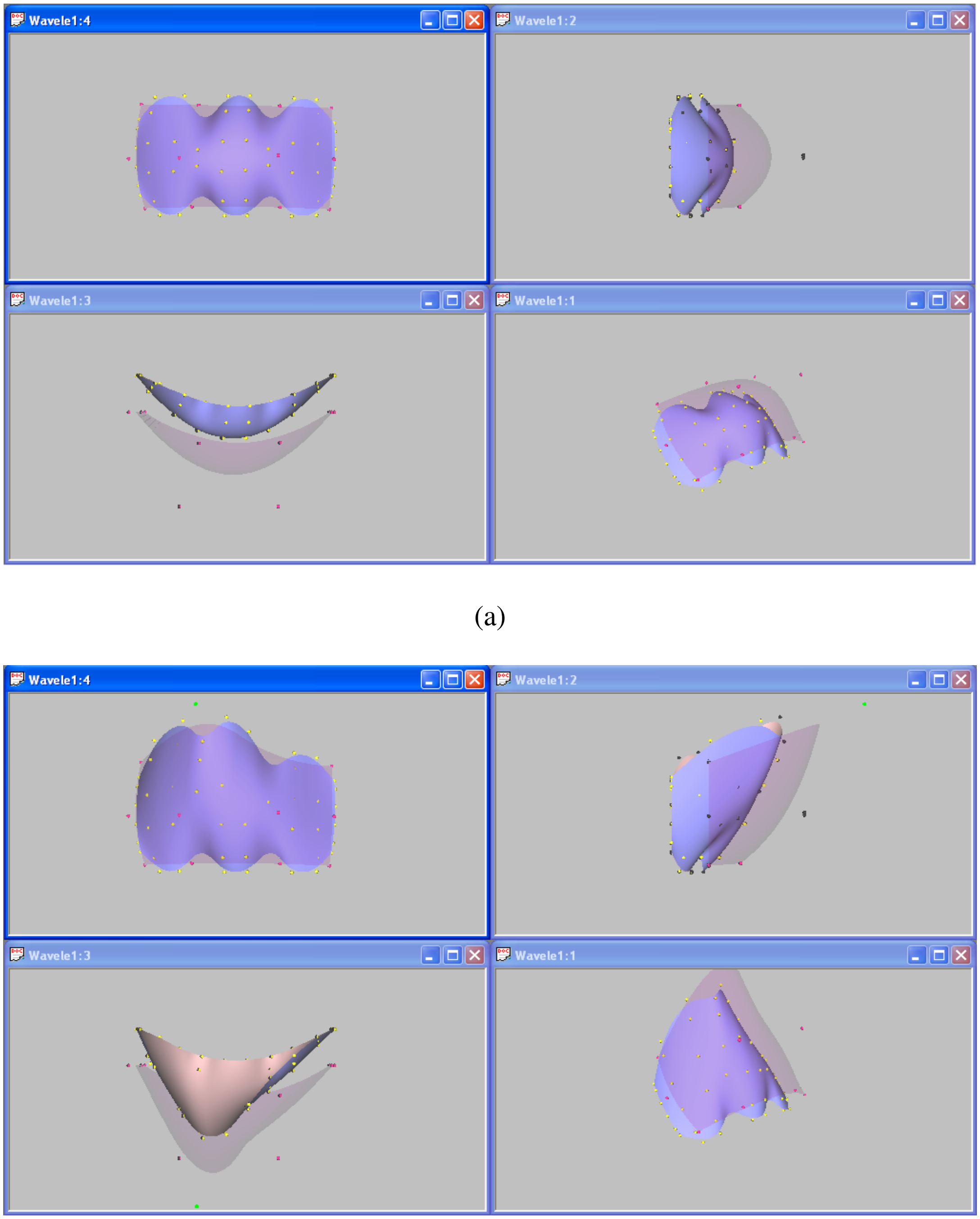


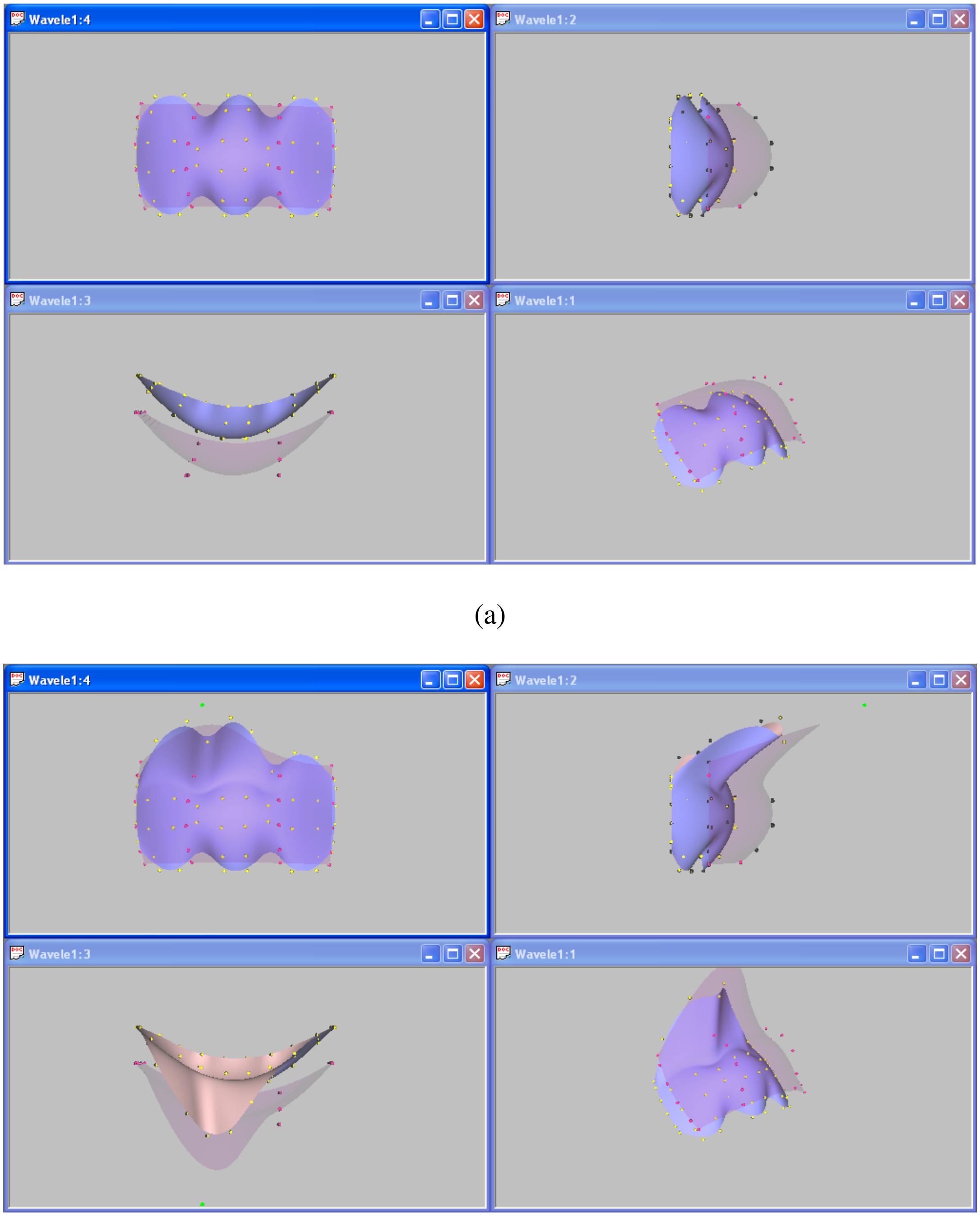
Figure 4.21 Overall sweep editing: there are four resolution levels ($i = 0,1,2,3$) in the u direction and three resolution levels ($j = 0,1,2$) in the v direction at the original surface; (a) the solid surface is the one at the highest level ($i = 3,\ j = 2$); the transparent surface is the one at lower level ($i = 0,\ j = 2$); (b) the overall sweep of the surface after editing one control point (green) at level ($i = 0,\ j = 2$).



(a)

(b)

Figure 4.22 Overall sweep editing: there are four resolution levels ($i = 0,1,2,3$) in the u direction and three resolution levels ($j = 0,1,2$) in the v direction at the original surface; (a) the solid surface is the one at the highest level ($i = 3, j = 2$); the transparent surface is the one at lower level ($i = 1, j = 0$); (b) the overall sweep of the surface after editing one control point (green) at level ($i = 1, j = 0$).



(a)

(b)

Figure 4.23 Overall sweep editing: there are four resolution levels ($i = 0, 1, 2, 3$) in the u direction and three resolution levels ($j = 0, 1, 2$) in the v direction at the original surface; (a) the solid surface is the one at the highest level ($i = 3,\ j = 2$); the transparent surface is the one at lower level ($i = 1,\ j = 2$); (b) the overall sweep of the surface after editing one control point (green) at level ($i = 1,\ j = 2$).

surface is refined to a higher function space where the required fine detail features can be achieved. Since the subdivision scheme leads to the B-spline nested spaces, B-spline wavelet decomposition can be applied to the refined surface. This can effectively manage the balance between local and global control for the B-spline surfaces.

The steps for performing surface detail editing are listed as follows:

Step 1: Subdivision to extend the resolution level in the u and v directions. The user determines the path to construct a high resolution level. Figure 4.24 shows an example of refining a surface first along the v direction, and then along the u direction.

$$\begin{tikzcd}
C^{m,n} & C^{m-1,n} \arrow[l, "\textit{subdivision}"'] & \cdots & C^{u+1,n} & C^{u,n} \arrow[l, "\textit{subdivision}"'] \\
&&&& C^{u,n-1} \arrow[u, "\textit{subdivision}"'] \\
&&&& \vdots \\
&&&& C^{u,v+1} \\
&&&& C^{u,v} \arrow[u, "\textit{subdivision}"']
\end{tikzcd}$$

Figure 4.24 Level refinement from resolution level (u, v) to resolution level (m, n).

Step 2: Detail editing at the highest resolution level, as shown in Figure 4.25.

Step 3: Decomposition for the surface. Since detail editing is executed at the highest resolution level, the surface needs to be decomposed to update all the low resolution levels. Figure 4.26 presents an example of decomposition for the edited surface.

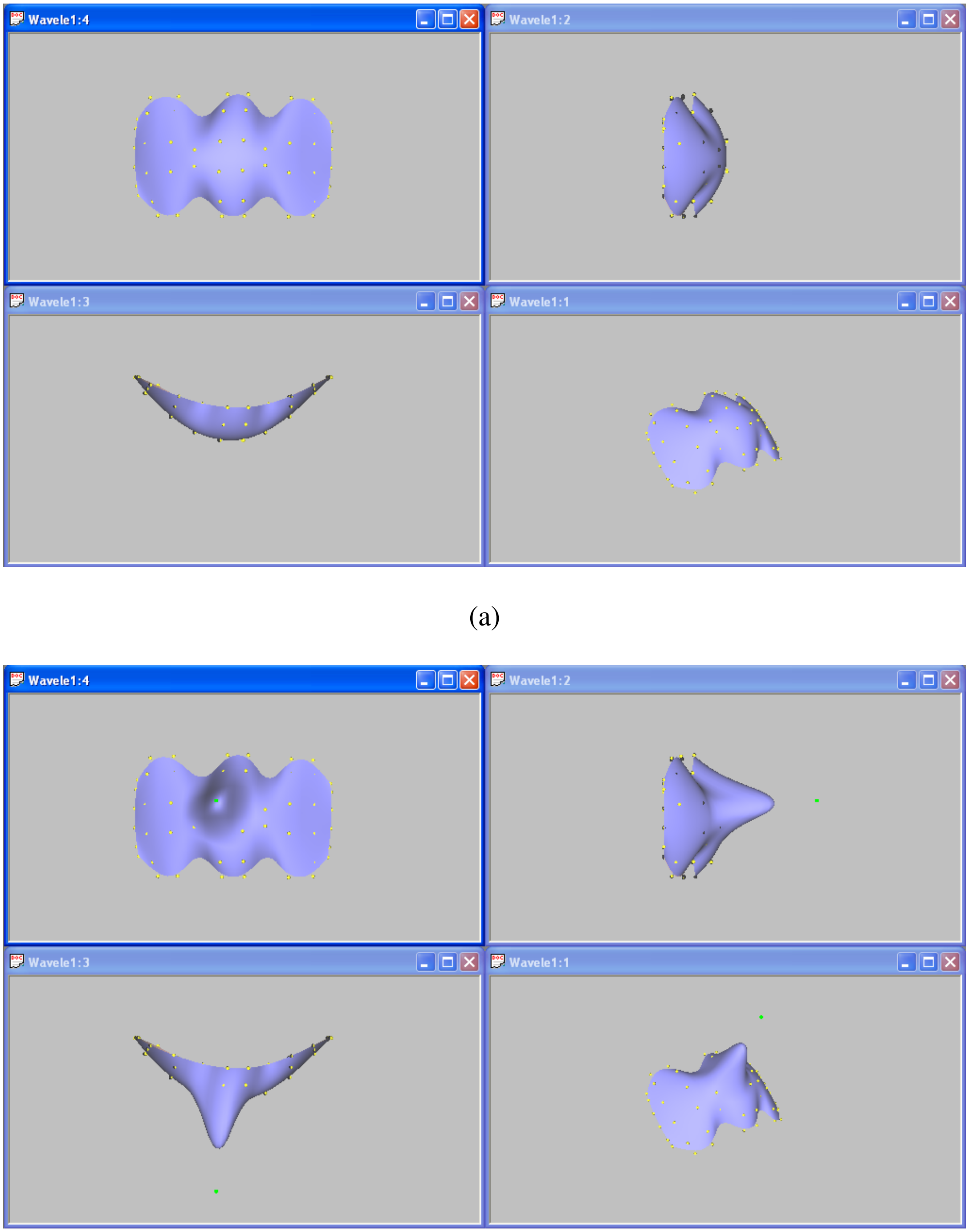
$$\begin{tikzcd}
\Delta C^{m,n} \arrow[d] \\
C^{m,n}
\end{tikzcd}$$

Figure 4.25 Detail editing at the highest resolution level (m, n).

$$\begin{tikzcd}
& \overline{D}^{m-1,n} & & \overline{D}^{m-2,n} & & & & \overline{D}^{0,n} & & \\
\overline{C}^{m,n} \arrow[ur] \arrow[rr] \arrow[dr] \arrow[dd] & & \overline{C}^{m-1,n} \arrow[ur] \arrow[rr] \arrow[dr] \arrow[dd] & & \overline{C}^{m-2,n} \arrow[dr] \arrow[dd] & \cdots & \overline{C}^{1,n} \arrow[ur] \arrow[rr] \arrow[dr] \arrow[dd] & & \overline{C}^{0,n} \arrow[dr] \arrow[dd] & \\
& \overline{D}^{m,n-1} & & \overline{D}^{m-1,n-1} & & \overline{D}^{m-2,n-1} & & \overline{D}^{1,n-1} & & \overline{D}^{0,n-1} \\
\overline{C}^{m,n-1} & & \overline{C}^{m-1,n-1} & & \overline{C}^{m-2,n-1} & & \overline{C}^{1,n-1} & & \overline{C}^{0,n-1} & \\
\vdots & & \vdots & & \vdots & & \vdots & & \vdots & \\
\overline{C}^{m,1} \arrow[dr] \arrow[dd] & & \overline{C}^{m-1,1} \arrow[dr] \arrow[dd] & & \overline{C}^{m-2,1} \arrow[dr] \arrow[dd] & & \overline{C}^{1,1} \arrow[dr] \arrow[dd] & & \overline{C}^{0,1} \arrow[dr] \arrow[dd] & \\
& \overline{D}^{m,0} & & \overline{D}^{m-1,0} & & \overline{D}^{m-2,0} & & \overline{D}^{1,0} & & \overline{D}^{0,0} \\
\overline{C}^{m,0} & & \overline{C}^{m-1,0} & & \overline{C}^{m-2,0} & & \overline{C}^{1,0} & & \overline{C}^{0,0} &
\end{tikzcd}$$

Figure 4.26 Decomposition for the edited surface.

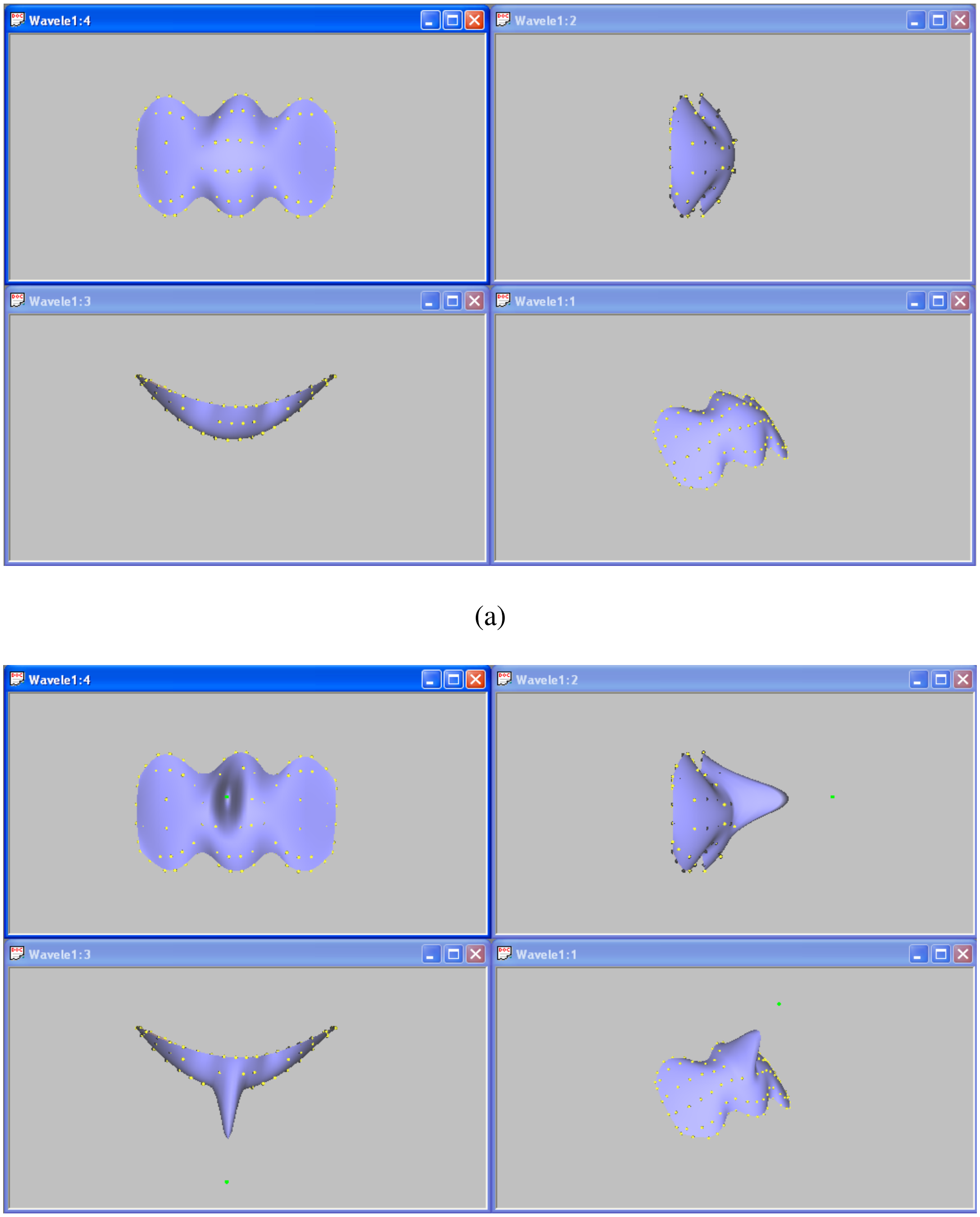
Figures 4.27 - 4.30 illustrate detail editing for a B-spline surface at different resolution levels. Four viewports are depicted in each picture: Front (top left), Left (top right), Top (bottom left), and Perspective (bottom right). The subdivision surface is shown in (a), and detail editing at the subdivision surface is shown in (b). Figure 4.31 shows an example of global control for the subdivision surface.



(a)

(b)

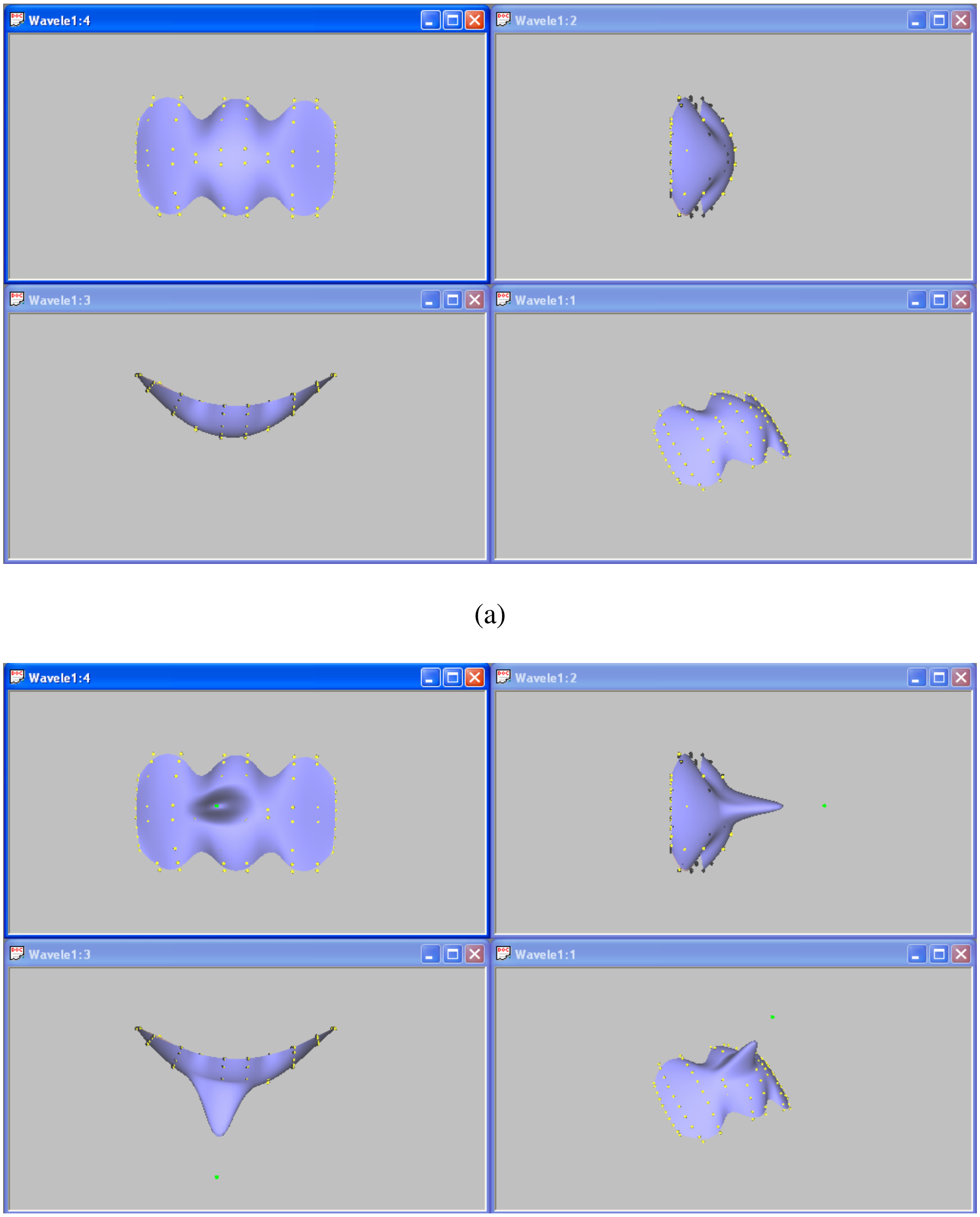
Figure 4.27 Detail editing: there are four resolution levels ($i = 0,1,2,3$) in the u direction and three resolution levels ($j = 0,1,2$) in the v direction at the original surface; (a) the solid surface is the one at the highest level ($i = 3,\ j = 2$); (b) detail editing after editing one control point (green) at level ($i = 3,\ j = 2$).



(a)

(b)

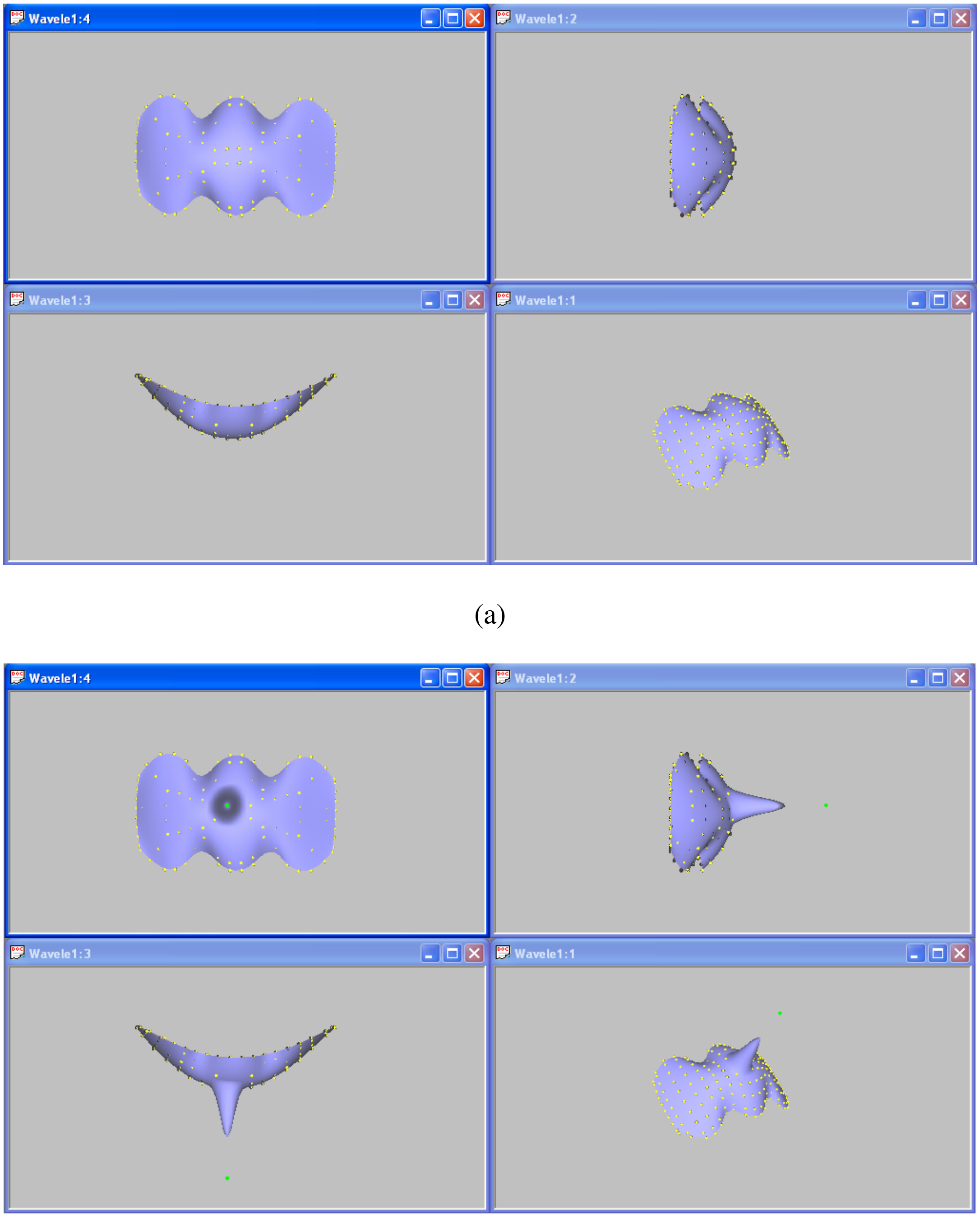
Figure 4.28 Detail editing: there are four resolution levels ($i = 0, 1, 2, 3$) in the u direction and three resolution levels ($j = 0, 1, 2$) in the v direction at the original surface; (a) the solid surface is the original surface after one step subdivision in the u direction; the surface is refined to next higher resolution level in the u direction ($i = 4,\ j = 2$); (b) detail editing after editing one control point (green) at level ($i = 4,\ j = 2$).



(a)

(b)

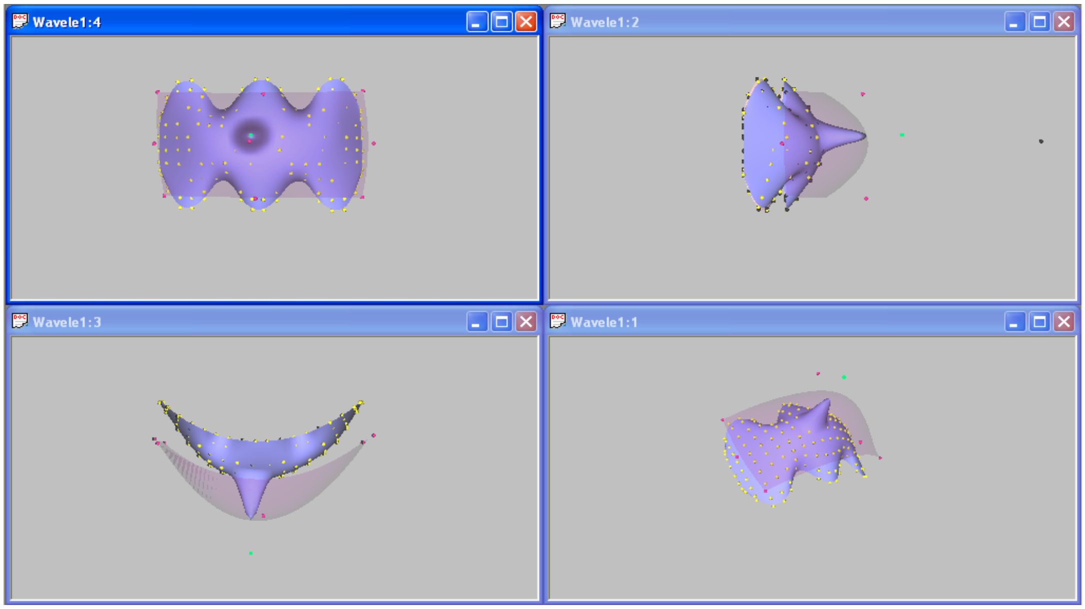
Figure 4.29 Detail editing: there are four resolution levels ($i = 0, 1, 2, 3$) in the u direction and three resolution levels ($j = 0, 1, 2$) in the v direction at the original surface; (a) the solid surface is the original surface after one step subdivision in the v direction; the surface is refined to next higher resolution level in the v direction ($i = 3,\ j = 3$); (b) detail editing after editing one control point (green) at level ($i = 3,\ j = 3$).



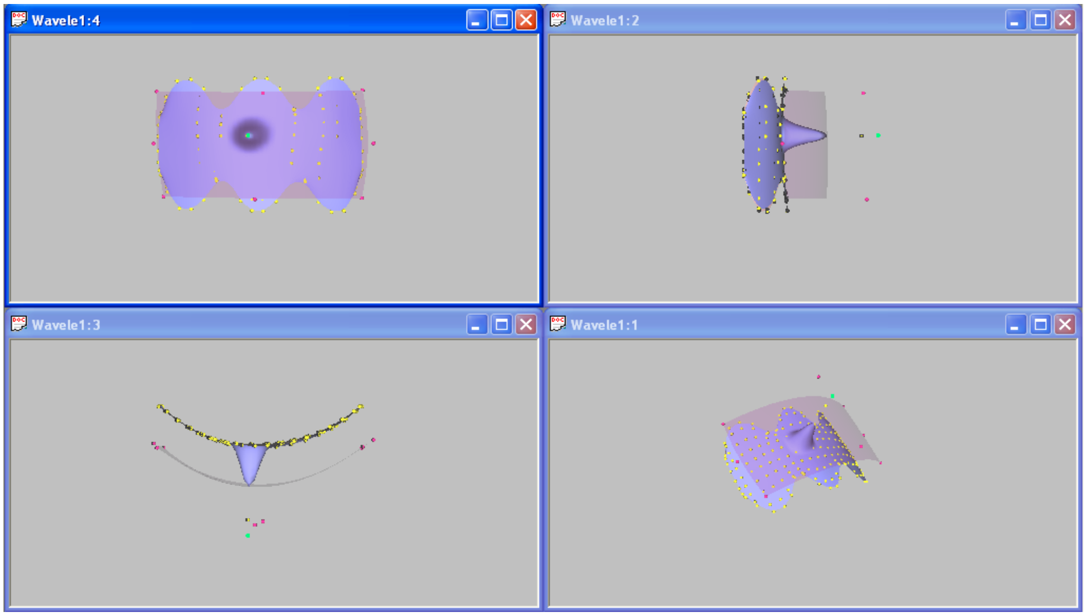
(a)

(b)

Figure 4.30 Detail editing: there are four resolution levels ($i = 0, 1, 2, 3$) in the u direction and three resolution levels ($j = 0, 1, 2$) in the v direction at the original surface; (a) the solid surface is the original surface after one step subdivision in the u and v directions; the surface is refined to next higher resolution level in the u and v directions ($i = 4$, $j = 3$); (b) detail editing after editing one control point (green) at level ($i = 4$, $j = 3$).



(a)



(b)

Figure 4.31 Global control for the subdivision surface: there are four resolution levels ($i = 0,1,2,3$) in the u direction and three resolution levels ($j = 0,1,2$) in the v direction at the original surface; the solid surface is the original surface after one step subdivision in the u and v directions; (a) detail editing after editing one control point (green) at level ($i = 4,\ j = 3$); (b) overall sweep editing after editing one control point (green) at level ($i = 0,\ j = 0$).

CHAPTER 5. MULTIRESOLUTION SURFACE EDITING WITH APPLICATION TO PRODUCT DESIGN

Overall sweep editing, detail blending, and detail orientation are a series of methods based on multiresolution analysis used to manipulate curves and surfaces. This chapter presents the application of these methods in product design and styling. The concept of overlays is additionally developed as a convenient and efficient method for superposing detail features on a surface.

5.1 Introduction

B-spline wavelets allow multiresolution representation of curves and surfaces [38]. This property can be used to significant advantage for algorithms in computer-aided design and manufacturing. In this section, application to product design and styling are presented.

Multiresolution surface representations have received considerable attention in recent years in the fields of geometric modeling, computer graphics, and visualization, primarily because they enable designing and editing of a surface at different levels of detail. In addition, the multiresolution approach brings with it other useful properties such as local and global level of details, efficient surface compression, progressively refinable reconstruction, and error bounds [1].

In this section, multiresolution methods that can edit the surface at multiple resolution levels are presented. Using these methods, the user can choose either to edit the detail features of the object, or to edit the overall sweep of the object while preserving the detail features. Furthermore, detail features may be selectively hidden and superposed for editing convenience. The addition of detail features can be made at the highest resolution level by control point manipulation, sweep deformation and error control, or using a feature library and overlays. The concept of feature library and the use of overlays in a multiresolution framework represent new methods. These methods can significantly improve the efficiency and quality of product design and styling.

The remainder of the chapter is organized as follows. In section 5.2 the methods of multiresolution editing for free-form product design and styling are developed. Section 5.3 illustrates the use of these editing methods. Conclusions are presented in section 5.4.

5.2 Free-form Surface Editing

Given a B-spline surface C^n and all of its low resolution versions $C^0,...,C^{n-1}$ and details $D^0,...,D^{n-1}$, multiresolution analysis allows for two very different kinds of surface editing. If some low resolution version C^j is modified and then incorporated back in with the details $D^j,...,D^{n-1}$, the overall sweep of the surface will be modified. On the other hand, if the set of details $D^j,D^{j+1},...,D^{n-1}$ are modified while leaving the low resolution representations $C^j,...,C^{n-1}$ unchanged, the fine grained character of the surface will be modified without affecting its overall sweep. These two types of editing are explored more fully below.

5.2.1 Overall Sweep Editing

Two types of methods to edit the overall sweep of a surface and add additional details to the surface are now explored.

5.2.1.1 Resolution Level Editing

To obtain the desired overall sweep, the low resolution versions $C^0,...,C^{n-1}$ are edited, and detail features are added at the highest resolution level.

Assume C^n to be the control points of the original B-spline surface $S(u,v)$. Let C^k, $k=0,...,n-1$, be the low resolution versions of C^n, and let ΔC^k be an edited version of C^k at resolution level k. From Equation 2.6, a new equation for computing the edited version of a surface may be developed through reconstruction as follows:

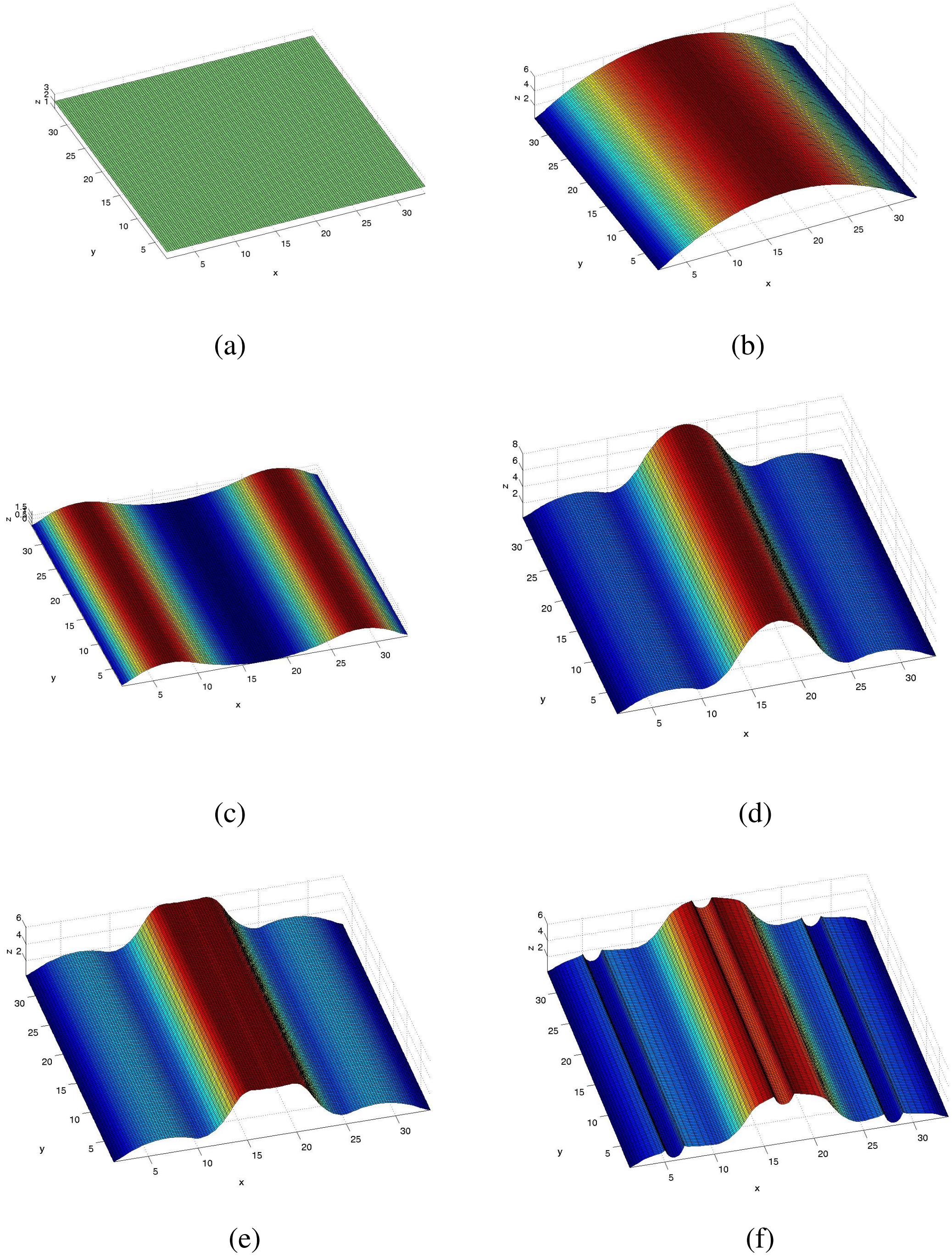
$$\overline{C}^n = C^n + \Delta C^n + P^n(\Delta C^{n-1} + P^{n-1}(... + P^2(\Delta C^1 + P^1 \Delta C^0)) \qquad (5.1)$$

where ΔC^n is the edited part at the highest resolution level. In this application, fine details are added at this level to define the detail features on the design surface. An example of this kind of editing is shown in Figure 5.1. 5.1(a) which shows the original model; 5.1(b)-5.1(e) shows the overall shapes after editing the central two control points at different resolution levels ($j = 1, 2, 3, 4$); 5.1(f) shows the detail features added at the highest resolution level ($j = 5$).

5.2.1.2 Error-Control Editing

The second method supported by overall sweep editing is *Error-Control* editing. In this method, a simple B-spline surface is defined first, and then fine details are added to the simple surface. To obtain the desired overall sweep of the surface, wavelet transforms are used to edit the surface at different resolution levels. Since this process can deform the shape of the details, an error control strategy is implemented to minimize the distortion.

The error control on the detail features is done using a least squares surface approximation approach. A series of data points on the referenced details are used to solve the least squares surface fitting problem.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.1 Resolution level editing: (a) the original model, consisting of 34×34 control points; (b) the overall sweep after editing the central two control points at level j = 1; (c) the overall sweep after editing the central two control points at level j = 2; (d) the overall sweep after editing the central two control points at level j = 3; (e) the overall sweep after editing the central two control points at level j = 4; (f) fine details added at the highest level (j = 5).

Assume that $s(u,v)=\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(u)N_{j,q}(v)\cdot c_{i,j}$ is a B-spline surface, and $\{Q_{k,l}\}$, $k=0,\cdots,n$ and $l=0,\cdots,m$, is a set of data points to be given. The objective function to approximate the data points in the least squares sense is:

$$\min\sum_{k=1}^{n-1}\sum_{l=1}^{m-1}\left|Q_{k,l}-s(\bar{u}_k,\bar{v}_l)\right|^2$$

with $Q_{0,0}=s(0,0)$, $Q_{n,0}=s(1,0)$, $Q_{0,m}=s(0,1)$, and $Q_{n,m}=s(1,1)$.

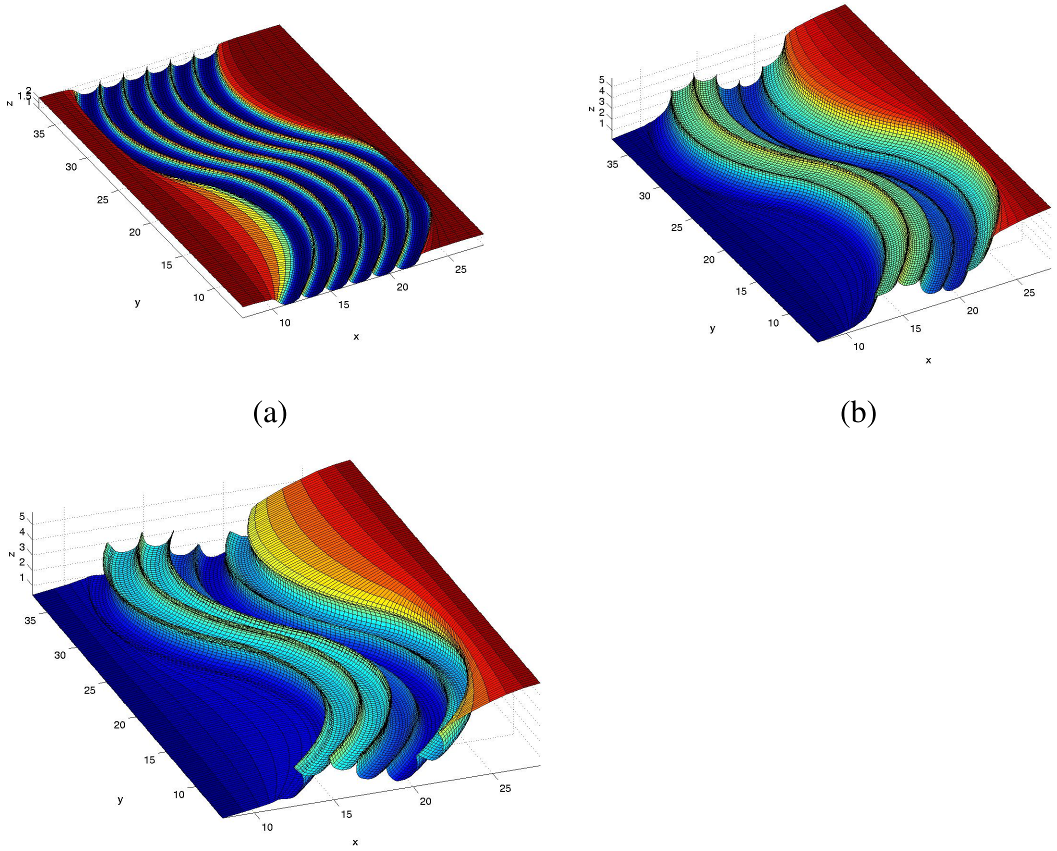
The solution for the least squares surface fitting is straightforward. First, curves are fit across the data in one direction, and then curves through the resulting control points are fit across the other direction. The detailed information for the algorithm can be found in [2]. Figure 5.2 shows the sequences resulting from this kind of editing method. Figure 5.2(c) shows the result after error control, and the relative error is within 10%.

5.2.2 Detail Editing

Another form of editing supported by the wavelet-based multiresolution surface is one of editing details on the surface without affecting its overall sweep. The details of the surface can be edited and then the desired features are incorporated into the surface via B-spline wavelet reconstruction. The following two methods are provided for editing details.

5.2.2.1 Detail Blending

The basic idea behind detail blending is to first filter out the different details using wavelet transforms, and then synthesize the desired detail features into a surface. In this application, simple surfaces with different detail features are defined first, then the detail



(a)

(b)

(c)

Figure 5.2 Error-Control editing: (a) the original model with details; (b) the overall sweep after resolution level editing; (c) the overall sweep after *Error-Control* editing; the relative error is less than 10%.

features are obtained from wavelet transforms, and finally the desired detail features are synthesized into a surface.

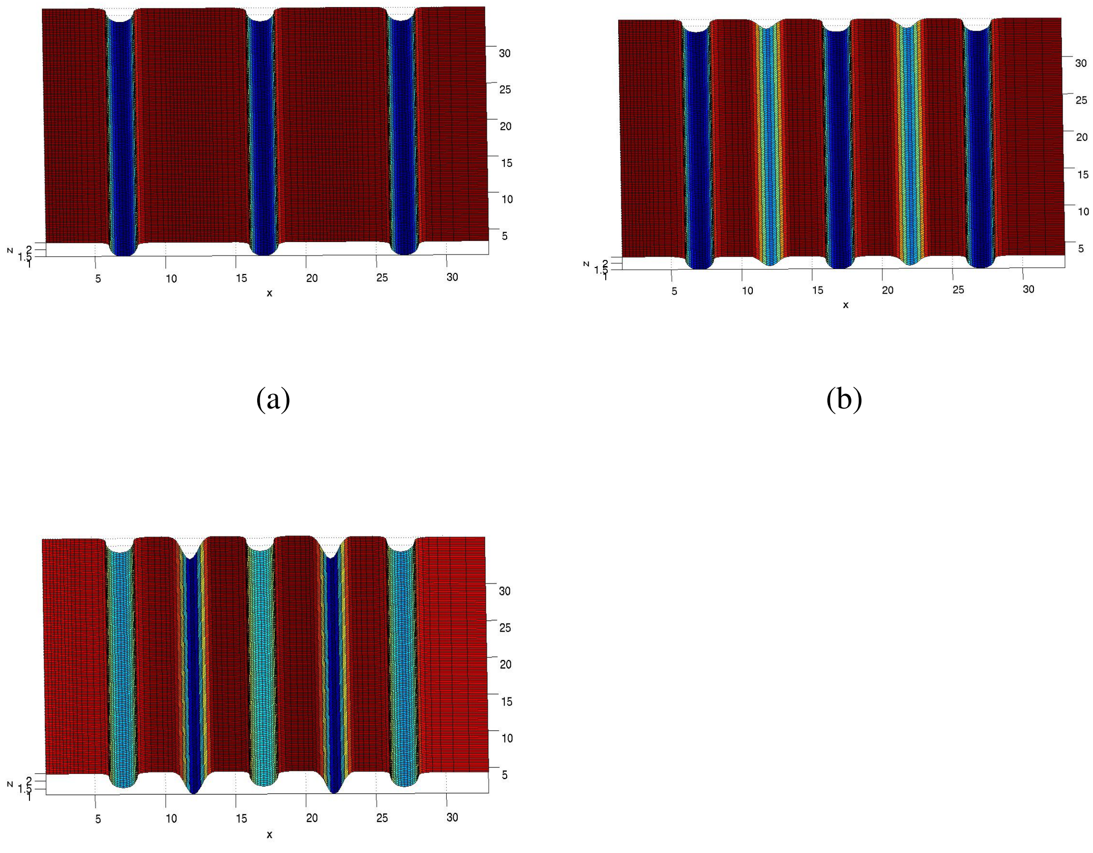
From Equation 2.6, an equation for calculating the edited version at resolution level $j+1$ can be developed:

$$C^{j+1} = P^{j+1}C^{j} + Q^{j+1}(D^{j} + \alpha_1 D_1^{j} + \cdots + \alpha_k D_k^{j}) \qquad (5.2)$$

where $D_1^j, \cdots, D_k^j$ are different details at resolution level j, and $\alpha_1, \cdots, \alpha_k$ are detail feature coefficients used to control the sizes of the details. Figure 5.3 illustrates the process of detail blending.

5.2.2.2 Detail Orientation

In practice, one often needs to edit part of a surface in order to add fine there. To achieve this objective, subdivision techniques can be employed to first construct an overlay on this



(a)

(b)

(c)

Figure 5.3 Detail blending: (a) the original model with details; (b) the model after adding another set of details; (c) the size of the details scaled by a factor of 2.

area, then add detail features on this overlay, and finally use the local reference frame of the overlay to orient the details.

From the discussion in section 4.1.2, the subdivision process can be expressed as

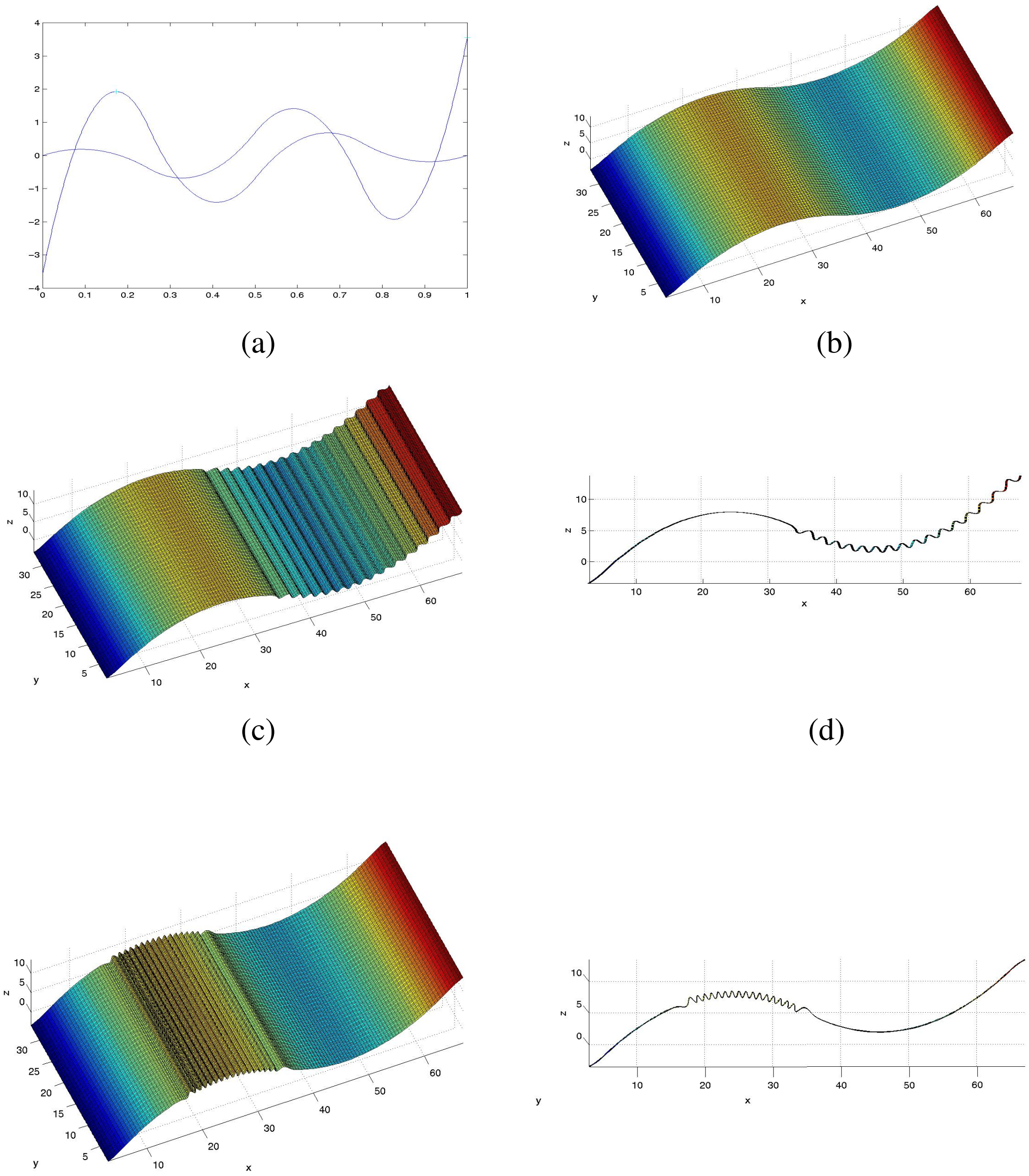
$$C^{j+1} = P^{j+1} C^{j} \tag{5.3}$$

where P^{j+1} is a rectangular constant matrix. Subdivision can recursively be applied on the edited area until the editing at a given resolution level on the overlay meets fine detail requirements.

A library of detail features have been developed to allow the user to choose different detail features. Since all detail features in the library are constructed along a straight axis, a local reference frame on the overlay is employed to orient the detail features. In this application the tangent and normal of the most influenced point of the overlay at a resolution level is used as the local reference frame of the detail features at the same level. The reference points correspond to the maximum values of the wavelets at that level. For example, Figure 5.4(a) shows the wavelet functions at the resolution level $j = 1$. The parameter positions corresponding to the maximum values of the wavelets are 0.173000 and 1.000000 respectively. These parameter positions are used as the reference points of the local reference frames, and the tangent and normal of these reference points are used to orient the added detail features at the same resolution level.

The detail orientation process can be expressed as:

$$\begin{bmatrix} D_{x'}^{j} \\ D_{y'}^{j} \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} D_{x}^{j} \\ D_{y}^{j} \end{bmatrix} \tag{5.4}$$



(a)

(b)

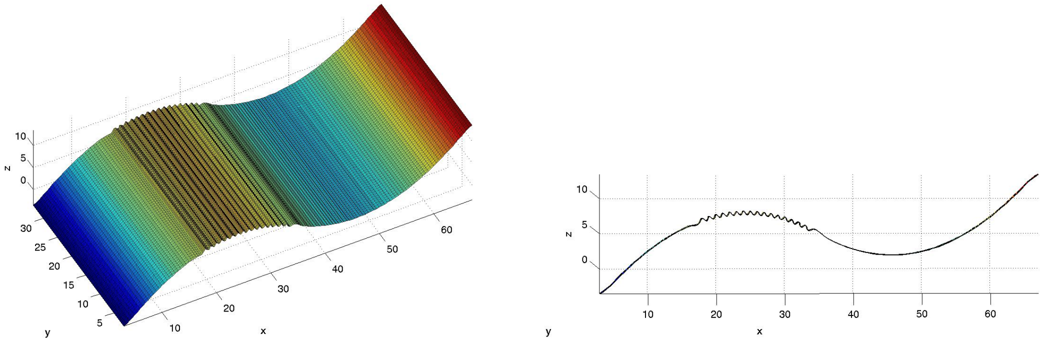
(c)

(d)

(e)

(f)

Figure 5.4 Detail orientation and overlay: (a) the wavelets at level j = 1; (b) the original model with 34×18 control points; (c) the model after adding one set of details on the right side using overlay and detail orientation; (d) the view of the model in the Y-axis direction; (e) the model after adding one set of details on the left side using overlay and detail orientation; (f) the view of the model in the Y-axis direction; (g) the model with the details scaled by a factor of 0.5; (h) the view of the model in the Y-axis direction.



(g)

(h)

Figure 5.4 (continued)

where D_x^j and D_y^j are the details in the fixed xy-orientation reference frame, $D_{x'}^j$ and $D_{y'}^j$ are the details oriented in the local reference frame, and α is the angle between the xy-orientation reference frame and the local reference frame.

The oriented details are added to the overlay in the detail blending method. Figure 5.4 shows an example using detail orientation.

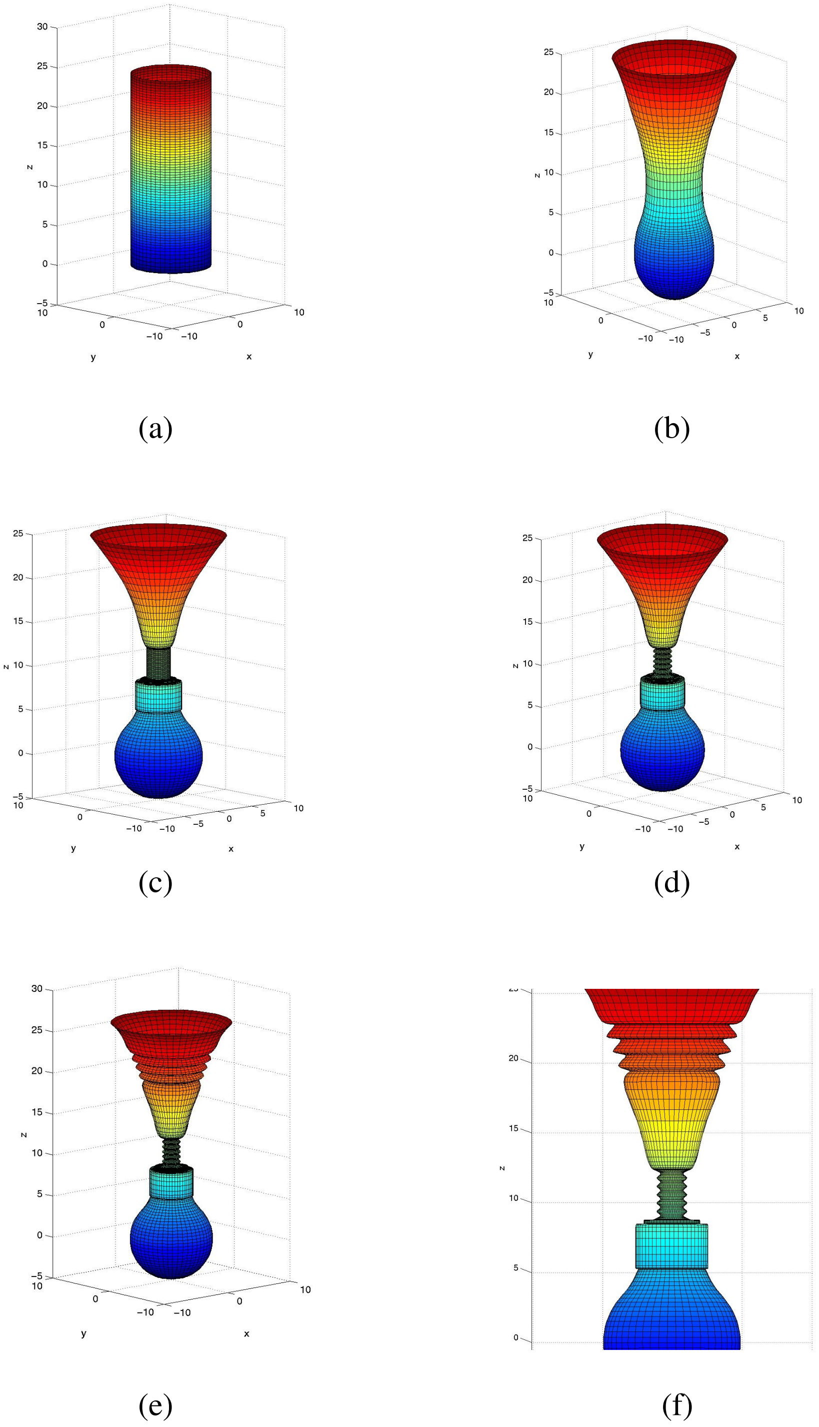
5.3 Results

The methods described in section 5.2 can be combined to create a complex model with detail features. In practice, when the model is complicated one method is often insufficient to realize product design and styling. The following example shows how to combine these methods to set up a complex model with detail features.

Figure 5.5 shows an example of a toy bicycle horn. Initially a bicubic tensor-product surface is set up to represent a simple cylinder. The total number of control points is 35×35, and there are six resolution levels in each direction. At the lowest resolution level, several control points are manipulated to edit the overall sweep of the model. Figure 5.5(b) shows the edited result after reconstruction. At the highest resolution level, the control points are edited to add detail features to the model, as shown in Figure 5.5(c). In Figure 5.5(d), the detail blending method is used to add one kind of detail feature on the model. The detail feature is obtained from the feature library with the size scaled to 1/3. In Figure 5.5(e), an overlay on the upper part of the model is constructed, and another kind of detail feature is obtained from the feature library with double the size of the original one, and finally detail orientation is applied to these features. Figure 5.5(f) shows the enlarged model viewed from the Y-axis direction.

5.4 Conclusions

In this chapter, a series of methods based on multiresolution analysis are described with applications to product design and styling. These methods include overall sweep editing and detail editing. A detail feature library is defined from which the user can select detail features as a convenient and interactive method for adding and editing details. The concept of overlay and detail orientation are used to apply these features. Since the method for orienting details with respect to tangent and normal properties are employed at a coarser resolution level, error and deformation in detail features will result. More accurate detail orientation methods may need to be developed in the future.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.5 A bicycle horn designed by using multiresolution surface editing: (a) a simple cylinder designed from a bicubic B-spline surface; (b) the model after resolution level editing at level j = 0; (c) the model after resolution level editing at level j = 5; (d) the model after adding one set of details using detail blending; (e) the model after adding another set of details using detail orientation; (f) the enlarged model viewed in the Y-axis direction.

CHAPTER 6. CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This dissertation introduces a new multiresolution approach to the creation of certain classes of free-form curves and surfaces that allows a designer to more easily manage the balance between local and global control. The techniques presented in this dissertation utilize a wavelet decomposition of a B-spline curve or surface to allow a designer to easily develop the basic shape, and then seamlessly switch to higher levels of detail to add additional definition. A surfacing kernel based on these wavelet-based B-splines is used as the basis for developing 3D curve and free-form surface modeling techniques. The algorithms described in the dissertation are implemented in a software system that is used to illustrate their utility and compare them to existing methods.

6.2 Future Work

Multiresolution analysis based on wavelets plays an increasing role in computer graphics and geometric modeling. Its utility as a powerful tool for efficiently representing functions at multiple levels of detail is expected to increase as its usefulness is further recognized. However, many issues are still open and should be explored before the full power of this representation can be realized. The following sub-sections discuss several areas where additional related work may be fruitful.

6.2.1 Multiresolution Texture Mapping

Multiresolution editing techniques can be utilized to simplify the assignment of texture coordinates for complex textures. In background modeling, several 2D images are used as backgrounds to construct a geometric model. This construction is only a coarse mapping. Delineating a set of features in an image will be the first step to refining the geometry. The positions of these features in texture space can be obtained by specifying the corresponding features in other images. The user could then specify the final positions that these features should take on the geometric model. Multiresolution editing techniques could be utilized to modify the geometry so that these texture features properly correspond to the desired positions on the geometry. This method will unify the modeling process and the texturing process into a single method, and is expected to avoid texture distortion and some of the complications of more conventional texture mapping approaches.

6.2.2 Multiresolution Editing for Rational B-spline Curves and Surfaces

In this dissertation, three-dimensional space is used, and x, y, and z coordinates of B-spline control points are manipulated to define the different levels of resolution. Some designers may wish to work with rational B-spline curves and surfaces to represent a number of specific curve and surface types, e.g. circles, ellipses, cylinder, cones, spheres, etc. Multiresolution B-spline representation can be extended to include rational B-spline curves and surfaces. Rational B-spline curves and surfaces use homogeneous coordinates to represent the curves and surfaces in three-dimensional space as polynomial curves and surfaces in four-dimensional space [2]. The weight w is used as an extra dimension, wherein any point in four-dimensional space is expressed as $P^w = (x, y, z, w)$. To apply multiresolution editing for ration B-spline curves and surfaces, the weight w should be included in Equation 2.4, Equation 2.5, and Equation 2.6. Wavelet decomposition and reconstruction for weights will follow the same steps as x, y, and z coordinates of control points. The B-spline wavelet transform approach is also easily extended to accommodate additional dimensions. It will involve appending another dependent variable to B-spline curves or surfaces.

APPENDIX. B-SPLINES

B-splines provide a unified, robust representation for free-form curves and surfaces. The following is a brief introduction to the B-spline curve and surface construction techniques used in this work. For a more complete presentation see references [2, 3, 4, 40, 41].

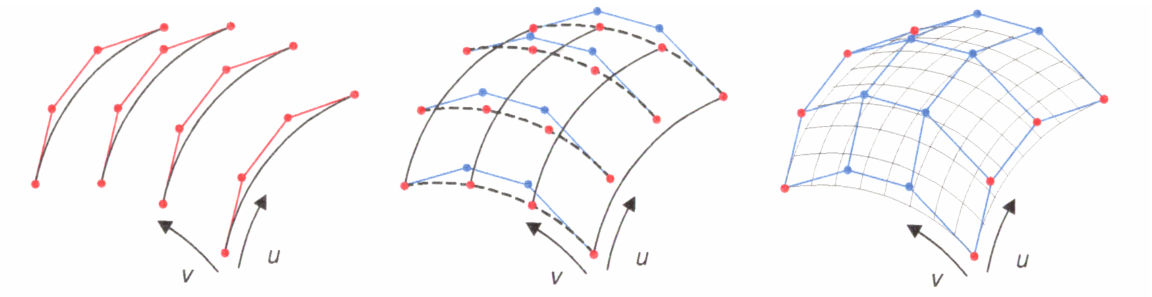
A.1 Surface Lofting

Surface lofting is a standard B-spline surface construction technique. Lofting, also known as skinning, is the process of blending a set of section curves to create a B-spline surface [2]. The blend direction, sometimes referred to as the longitudinal direction, is denoted as the v direction.

Lofting is defined as follows. Let

$$f_k(u) = \sum_{i=0}^{n} N_{i,d}(u) \cdot c_{i,k} \quad k = 0, \cdots, m \tag{A.1}$$

be the section curves. Each section curve must have the same knot vector U and common degree d. In addition, each section curve is assigned a parametric value $\bar{v}_k$ based on the averaging of the control point spacing. The control points of the section curves are then interpolated in the v direction, yielding the control points $c_{i,j}$ of the lofting surface. Figure A.1a shows four B-spline curves and their control points. Four v-directional interpolations are then executed, as shown in Figure A.1b. The resulting control points in Figure A.1b are actually the control points for a B-spline surface that interpolates the original four curves. The surface is shown in Figure A.1c.



(a) original curves (b) interpolated control points (c) resulting surface

Figure A.1 Surface lofting.

A.2 Parameterization

Given a set of data points $\{Q_k\}$, $k = 0, 1, \cdots, n$, parameterization is the process for assigning parametric values $\bar{u}_k$ to the points Q_k. There are three common methods for parameterization: *equally spaced*, *chord length*, and *centripetal* [2].

In the *equally spaced* method, the parameters are defined as

$$\bar{u}_0 = 0 \qquad \bar{u}_n = 1$$

$$\bar{u}_k = \frac{k}{n} \qquad k = 1, \cdots, n-1 \tag{A.2}$$

To assign parameters using the *chord length* method, first calculate the total chord length

$$d = \sum_{k=1}^{n} |Q_k - Q_{k-1}| \tag{A.3}$$

then assign the parameters as follows

$$\bar{u}_0 = 0 \qquad \bar{u}_n = 1$$

$$\bar{u}_k = \bar{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} \qquad k = 1, \cdots, n-1 \tag{A.4}$$

The *chord length* is a widely used method, and it produces a good approximation for uniform parameterization.

The *centripetal* method is a newer method. Let

$$d = \sum_{k=1}^{n} \sqrt{|Q_k - Q_{k-1}|} \tag{A.5}$$

The parameters of the *centripetal* method are defined as

$$\bar{u}_0 = 0 \qquad \bar{u}_n = 1$$

$$\bar{u}_k = \bar{u}_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{d} \qquad k = 1, \cdots, n-1 \tag{A.6}$$

The *centripetal* method gives a better result when the spacing of the data points varies.

A.3 Global Interpolation

Global interpolation is a process of calculating a B-spline curve that will interpolate a set of data points $\{Q_k\}$, $k = 0, 1, \cdots, n$, at the designated parametric values $\{\bar{u}_k\}$ [2, 3, 44]. The $(n+1) \times (n+1)$ system of linear equations to be solved is

$$Q_k = f(\bar{u}_k) = \sum_{i=0}^{n} N_{i,d}(\bar{u}_k) \cdot c_i \qquad k = 0, \cdots, n \tag{A.7}$$

The $(m+1)$ knots are calculated using the following technique of averaging

$$u_0 = \cdots = u_d = 0 \qquad u_{m-d} = \cdots = u_m = 1$$

$$u_{j+d} = \frac{1}{d} \sum_{i=j}^{j+d-1} \bar{u}_i \qquad j = i, \cdots, n-d \tag{A.8}$$

Equation A.7 can be rewritten in matrix notation as follows

$$\begin{bmatrix} Q_0 \\ Q_1 \\ \vdots \\ Q_{n-1} \\ Qn \end{bmatrix} = \begin{bmatrix} N_{0,d}(u_0) & N_{1,d}(u_0) & \cdots & N_{n-1,d}(u_0) & N_{n,d}(u_0) \\ N_{0,d}(u_1) & N_{1,d}(u_1) & \cdots & N_{n-1,d}(u_1) & N_{n,d}(u_1) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ N_{0,d}(u_{n-1}) & N_{1,d}(u_{n-1}) & \cdots & N_{n-1,d}(u_{n-1}) & N_{n,d}(u_{n-1}) \\ N_{0,d}(u_n) & N_{1,d}(u_n) & \cdots & N_{n-1,d}(u_n) & N_{n,d}(u_n) \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix}$$

which can be solved using LU decomposition.

REFERENCES

[1] Dreger, A., Gross, M. H., and Schlegel, J., Multiresolution Triangular B-spline Surfaces, *Computer Graphics International 1998*, 1998, pp. 166-177.

[2] Piegl, L. and Tiller, W., *The NURBS Book*, Springer Verlag, Germany, 1997.

[3] Bohm, W., Farin, G., and Kahmann, J., A Survey of Curve and Surface Methods in CAGD, *Computer Aided Geometric Design*, Vol. 1, No. 1, 1984, pp. 1-60.

[4] De Boor, C., On calculating with B-splines, *Journal of Approximation Theory*, 1972, pp. 50-62.

[5] Barr, A. H., Global and Local Deformation of Solid Primitives, *Computer Graphics*, Vol. 18, No. 3, 1984, pp. 21-30.

[6] Cobb, E. S., Design of Sculptured Surfaces Using the B-spline Representation, Ph.D. Thesis, University of Utah, 1984.

[7] Sederberg, T. and Parry, S., Free-Form Deformation of Solid Geometric Models, *Computer Graphics*, Vol. 20, No. 4, 1986, pp. 151-160.

[8] Coquillart, S., Extended Free-Form Deformations: A Sculpturing Tool for 3D Geometric Modeling, *Computer Graphics*, Vol. 24, No. 4, 1990, pp. 187-196.

[9] Forsey, D. R. and Bartels, R. H., Hierarchical B-spline Refinement, *Computer Graphics,* Vol. 22, No. 4, 1988, pp. 205-212.

[10] Forsey, D. R. and Wong, D., Multiresolution Surface Reconstruction for Hierarchical B-splines, Tech. rep., University of British Columbia, 1995.

[11] Yvart, A. and Hahmann, S., Hierarchical Triangular Splines, *ACM Transactions on Graphics*, Vol. 4, No. 24, 2005, pp. 1374-1391.

[12] Hahmann, S. and Bonneau G.-P., Polynomial Surfaces Interpolating Arbitrary Triangulations, *IEEE Transactions on Visualization and Computer Graphics 9*, 1, 2003, pp. 99-109.

[13] Finkelstein, A. and Salesin, D. H., Multiresolution Curves, *Computer Graphics*, Vol. 28, 1994, pp. 261-268.

[14] Gortler, S. J. and Cohen, M. F., Hierarchical and Variational Geometric Modeling with Wavelets, *In Proceeding Symposium on Interactive 3D Graphics*, May 1995, pp. 35-42.

[15] Lounsbery, M., DeRose, T. D., and Warren, J., Multiresolution Analysis for Surfaces of Arbitrary Topological Type, *ACM Transactions on Graphics*, Vol. 16, No. 1, 1997, pp. 34-73.

[16] Schroder, P. and Sweldens, W., Spherical Wavelets: Efficiently Representing Functions on the Sphere, *Computer Graphics*, Vol. 29, 1995, pp. 161-172.

[17] Zorin, D., Schroder, P., and Sweldens, W., Interpolation Subdivision for Meshes with Arbitrary Topology, *Computer Graphics*, Vol. 30, 1996, pp. 189-192.

[18] Zorin, D., Schroder, P., and Sweldens, W., Interactive Multiresolution Mesh Editing, *Computer Graphics*, Vol. 31, 1997, pp. 259-268.

[19] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W., Multiresolution Analysis of Arbitrary Meshes, *Computer Graphics Proceedings*, Annual Conference Series, 1995, pp. 173-182.

[20] Bonneau, G.-P., Elber, G., Hahmann, S., and Sauvage, B., Multiresolution Analysis, *Shape Analysis and Structuring*, Springer Verlag, 2008, pp. 83-114.

[21] Cheutet, V., Daniel, M., Hahmann, S., La Greca, R., Leon, J.-C., Machlet, R., Menegaux, D., and Sauvage, B., *Constraint Modeling for Curves and Surfaces in CAGD*, International Journal of Shape Modeling, 2007.

[22] Elber, G., Multiresolution Curve Editing with Linear Constraints, *ACM Symposium on Solid and Physical Modeling*, June 2001, pp. 109-119.

[23] Hahmann, S., Sauvage, B., and Bonneau, G.-P., Area Preserving Deformation of Multiresolution Curves, *Computer Aided Geometric Design*, Vol. 22, No. 4, 2005, pp. 349-367.

[24] Sauvage, B., Hahmann, S., Bonneau, G.-P., and Elber, G., Detail Preserving Deformation of B-spline Surfaces with Volume Constraint, *Computer Aided Geometric Design*, 2007.

[25] Sauvage, B., Hahmann, S., and Bonneau, G.-P., Length Preserving Multiresolution Editing of Curves, *Computing*, Vol. 72, 2004, pp. 161-170.

[26] Sauvage, B., Hahmann, S., and Bonneau, G.-P., Length Constrained Multiresolution Deformation for Surface Wrinkling, *Shape Modeling and Applications*, June 2006, pp. 113-121.

[27] Stollnitz, E. J., DeRose, T. D., and Salesin, D. H., Wavelets for Computer Graphics: A Primer, Part 1, *IEEE Computer Graphics and Applications*, Vol. 15, No. 3, May 1995, pp. 76-84.

[28] Mallat, S., A Theory for Multiresolution Signal Decomposition: The Wavelet Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, July 1989, pp. 674-693.

[29] Meyer, Y., Wavelets and Operators, *Cambridge Studies in Advanced Math*, Vol. 37, Cambridge University Press, 1993.

[30] Jawerth, B. and Sweldens, W., An Overview of Wavelet Based Multiresolution Analyses, SIAM Review, Vol. 36, No. 3, 1994, pp. 377-412.

[31] Quak, E. and Weyrich, N., Decomposition and Reconstruction Algorithms for Spline Wavelets on a Bounded Interval, *Applied and Computational Harmonic Analysis*, Vol. 1, No. 3, June 1994, pp. 217-231.

[32] Daubechies, I., Orthonormal bases of compactly supported wavelets, *Communications on Pure and Applied Mathematics*, Vol. 41, No. 7, 1988, pp. 909-996.

[33] Chui, C. K., *An Introduction to Wavelets*, Academic Press, Boston, 1992.

[34] Chui, C. K. and Quak, E. Wavelets on a bounded interval, *Numerical Methods of Approximation Theory*, Vol. 9, 1992, pp. 53-75.

[35] Sweldens, W., The lifting Scheme: A Custom-Design Construction of Biorthogonal Wavelets, *Applied and Computational Harmonic Analysis*, Vol. 3, No. 2, 1996, pp. 186-200.

[36] Stollnitz, E. J., DeRose, T. D., and Salesin, D. H., Wavelets for Computer Graphics: A Primer, Part 2, *IEEE Computer Graphics and Applications*, Vol. 15, No. 4, July 1995, pp. 75-85.

[37] Samavati, F. F. and Bartels, R. H., Multiresolution Curve and Surface Representation: Reverse Subdivision Rules by Least-Square Data Fitting, *Computer Graphic Forum*, Vol. 18, No. 2, 1999, pp. 97-119.

[38] Olsen, L., Samavati, F. F., and Bartels, R. H., Multiresolution for Curves and Surfaces Based on Constraining Wavelets, *Computers and Graphics*, Vol. 31, No. 3, June 2007, pp. 449-462.

[39] Boehm, W. and Prautzsch, H., The Insertion Algorithm, *CAD*, Vol. 17, No. 2, 1985, pp. 58-59.

[40] Piegl, L., On NURBS: A survey, *IEEE Computer Graphics and Applications*, Vol. 11, No. 1, 1991, pp. 55-71.

[41] Gordon, W. J. and Riesenfeld, R. F., B-spline Curves and Surfaces, *Computer Aided Geometric Design*, 1974, pp. 95-126.