Wavelets for Computer Graphics: A Primer
Part 2[†]

Eric J. Stollnitz *Tony D. DeRose* *David H. Salesin*

University of Washington

1 Introduction

Wavelets are a mathematical tool for hierarchically decomposing functions. They allow a function to be described in terms of a coarse overall shape, plus details that range from broad to narrow. Regardless of whether the function of interest is an image, a curve, or a surface, wavelets provide an elegant technique for representing the levels of detail present.

In Part 1 of this primer we discussed the simple case of Haar wavelets in one and two dimensions, and showed how they can be used for image compression. In Part 2, we present the mathematical theory of multiresolution analysis, then develop bounded-interval spline wavelets and describe their use in multiresolution curve and surface editing.

2 Multiresolution analysis

The Haar wavelets we discussed in Part 1 are just one of many bases that can be used to treat functions in a hierarchical fashion. In this section, we develop a mathematical framework known as *multiresolution analysis* for studying wavelets [2, 11]. Our examples will continue to focus on the Haar basis, but the more general mathematical notation used here will come in handy for discussing other wavelet bases in later sections.

Multiresolution analysis relies on many results from linear algebra. Some readers may wish to consult the appendix in Part 1 for a brief review.

As discussed in Part 1, the starting point for multiresolution analysis is a nested set of vector spaces

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

As j increases, the resolution of functions in V^j increases. The basis functions for the space V^j are known as *scaling functions.*

The next step in multiresolution analysis is to define *wavelet spaces.* For each j, we define W^j as the *orthogonal complement* of V^j in V^{j+1}. This means that W^j includes all the functions in V^{j+1} that are orthogonal to all those in V^j under some chosen inner product. The functions we choose as a basis for W^j are called *wavelets.*

2.1 A matrix formulation for refinement

The rest of our discussion of multiresolution analysis will focus on wavelets defined on a bounded domain, although we will also refer to wavelets on the unbounded real line wherever appropriate. In the bounded case, each space V^j has a finite basis, allowing us to use matrix notation in much of what follows, as did Lounsbery *et al.* [10] and Quak and Weyrich [13].

It is often convenient to put the different scaling functions $\phi_i^j(x)$ for a given level j together into a single row matrix,

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

where m^j is the dimension of V^j. We can do the same for the wavelets:

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

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

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

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

In other words, each scaling function at level $j-1$ must be expressible as a linear combination of "finer" scaling functions at level j. Note that since V^j and V^{j-1} have dimensions m^j and m^{j-1}, respectively, P^j is an $m^j \times m^{j-1}$ matrix (taller than it is wide).

Since the wavelet space W^{j-1} is by definition also a subspace of V^j, we can write the wavelets $\Psi^{j-1}(x)$ as linear combinations of the scaling functions $\Phi^j(x)$. This means there is an $m^j \times n^{j-1}$ matrix of constants Q^j satisfying

$$\Psi^{j-1}(x) = \Phi^j(x)\, Q^j. \tag{2}$$

Example: In the Haar basis, at a particular level j there are $m^j = 2^j$ scaling functions and $n^j = 2^j$ wavelets. Thus, there must be refinement matrices describing how the two scaling functions in V^1 and the two wavelets in W^1 can be made from the four scaling functions in V^2:

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

Remark: In the case of wavelets constructed on the unbounded real line, the columns of P^j are shifted versions of one another, as are the columns of Q^j. One column therefore characterizes each matrix, so P^j and Q^j are completely determined by sequences $(\ldots, p_{-1}, p_0, p_1, \ldots)$ and $(\ldots, q_{-1}, q_0, q_1, \ldots)$, which also do not depend on j. Equations (1) and (2) therefore often appear in the literature as expressions of the form

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

$$\psi(x) = \sum_i q_i\, \phi(2x - i).$$

[†] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for computer graphics: A primer, part 2. *IEEE Computer Graphics and Applications*, 15(4):75–85, July 1995.

These equations are referred to as *two-scale relations* for scaling functions and wavelets, respectively.

Note that equations (1) and (2) can be expressed as a single equation using block-matrix notation:

$$\left[\, \Phi^{j-1} \;\middle|\; \Psi^{j-1} \,\right] \;=\; \Phi^{j} \left[\, P^{j} \;\middle|\; Q^{j} \,\right]. \tag{3}$$

Example: Substituting the matrices from the previous example into Equation (3) along with the appropriate basis functions gives

$$[\phi_0^1 \;\; \phi_1^1 \;\; \psi_0^1 \;\; \psi_1^1] \;=\; [\phi_0^2 \;\; \phi_1^2 \;\; \phi_2^2 \;\; \phi_3^2] \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

It is important to realize that once we have chosen scaling functions and their refinement matrices P^j, the wavelet matrices Q^j are somewhat constrained (though not completely determined). In fact, since all functions in $\Phi^{j-1}(x)$ must be orthogonal to all functions in $\Psi^{j-1}(x)$, we know $\langle \phi_k^{j-1} \mid \psi_\ell^{j-1} \rangle = 0$ for all k and ℓ.

To deal with all these inner products simultaneously, let's define some new notation for a matrix of inner products. We will denote by $[\langle \Phi^{j-1} \mid \Psi^{j-1} \rangle]$ the matrix whose (k, ℓ) entry is $\langle \phi_k^{j-1} \mid \psi_\ell^{j-1} \rangle$. Armed with this notation, we can rewrite the orthogonality condition on the wavelets as

$$[\langle \Phi^{j-1} \mid \Psi^{j-1} \rangle] \;=\; \boldsymbol{0}. \tag{4}$$

Substituting Equation (2) into Equation (4) yields

$$[\langle \Phi^{j-1} \mid \Phi^{j} \rangle]\, Q^j \;=\; \boldsymbol{0}. \tag{5}$$

A matrix equation with a right-hand side of zero like this one is known as a *homogeneous* system of equations. The set of all possible solutions is called the *null space* of $[\langle \Phi^{j-1} \mid \Phi^{j} \rangle]$, and the columns of Q^j must form a basis for this space. There are a multitude of bases for the null space of a matrix, implying that there are many different wavelet bases for a given wavelet space W^j. Ordinarily, we uniquely determine the Q^j matrices by imposing further constraints in addition to the orthogonality requirement given above. For example, the Haar wavelet matrices can be found by requiring the least number of consecutive nonzero entries in each column.

The literature on wavelets includes various terminologies for orthogonality. Some authors refer to a collection of functions that are orthogonal to scaling functions but not to each other as *pre-wavelets*, reserving the term "wavelets" for functions that are orthogonal to each other as well. Another common approach is to differentiate between an *orthogonal wavelet basis*, in which all functions are mutually orthogonal, and a *semi-orthogonal wavelet basis*, in which the wavelets are orthogonal to the scaling functions but not to each other. The Haar basis is an example of an orthogonal wavelet basis, while the spline wavelets we will describe in Section 3 are examples of semi-orthogonal wavelet bases.

Finally, it is sometimes desirable to define wavelets that are not quite orthogonal to scaling functions in order to have wavelets with small supports. This last alternative might be termed a *non-orthogonal wavelet basis*, and we will mention an example when we describe multiresolution surfaces in Section 4.3.

2.2 The filter bank

The previous section showed how scaling functions and wavelets could be related by matrices. In this section, we show how matrix notation can also be used for the decomposition process outlined in Section 2.1 of Part 1.

Consider a function in some approximation space V^j. Let's assume we have the coefficients of this function in terms of some scaling function basis. We can write these coefficients as a column matrix of values $C^j = [c_0^j \;\cdots\; c_{m^j-1}^j]^{\mathrm{T}}$. The coefficients c_i^j could, for example, be thought of as pixel colors, or alternatively, as the x- or y-coordinates of a curve's control points in $\mathbb{R}^2$.

Suppose we wish to create a low-resolution version C^{j-1} of C^j with a smaller number of coefficients m^{j-1}. The standard approach for creating the m^{j-1} values of C^{j-1} is to use some form of linear filtering and down-sampling on the m^j entries of C^j. This process can be expressed as a matrix equation

$$C^{j-1} \;=\; A^j\, C^j \tag{6}$$

where A^j is an $m^{j-1} \times m^j$ matrix of constants (wider than it is tall).

Since C^{j-1} contains fewer entries than C^j, this filtering process clearly loses some amount of detail. For many choices of A^j, it is possible to capture the lost detail as another column matrix D^{j-1}, computed by

$$D^{j-1} \;=\; B^j\, C^j \tag{7}$$

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

If A^j and B^j are chosen appropriately, then the original coefficients C^j can be recovered from C^{j-1} and D^{j-1} by using the matrices P^j and Q^j from the previous section:

$$C^j \;=\; P^j\, C^{j-1} \;+\; Q^j\, D^{j-1}. \tag{8}$$

Recovering C^j from C^{j-1} and D^{j-1} is called *synthesis* or *reconstruction.* In this context, P^j and Q^j are called *synthesis filters.*

Example: In the unnormalized Haar basis, the matrices A^2 and B^2 are given by:

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

$$B^2 \;=\; \frac{1}{2} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

These matrices represent the averaging and differencing operations described in Section 2.1 of Part 1.

Remark: Once again, the matrices for wavelets constructed on the unbounded real line have a simple structure: The rows of A^j are shifted versions of each other, as are the rows of B^j. The analysis Equations (6) and (7) often appear in the literature as

$$c_k^{j-1} \;=\; \sum_\ell a_{\ell-2k}\, c_\ell^j$$

$$d_k^{j-1} \;=\; \sum_\ell b_{\ell-2k}\, c_\ell^j$$

where the sequences $(\ldots, a_{-1}, a_0, a_1, \ldots)$ and $(\ldots, b_{-1}, b_0, b_1, \ldots)$ are the entries in a row of A^j and B^j, respectively. Similarly, Equation (8) for reconstruction often appears as

$$c_k^j \;=\; \sum_\ell \left(p_{k-2\ell}\, c_\ell^{j-1} + q_{k-2\ell}\, d_\ell^{j-1} \right).$$

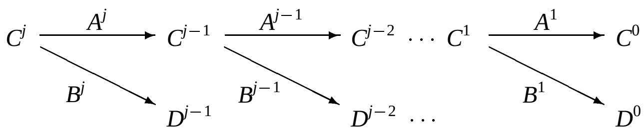

Figure 1 The filter bank.

Note that the procedure for splitting C^j into a low-resolution part C^{j-1} and a detail part D^{j-1} 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 $C^0, \ldots, C^{j-1}$ and details $D^0, \ldots, D^{j-1}$, as shown in Figure 1. This recursive process is known as a *filter bank.*

Since the original coefficients C^j can be recovered from the sequence C^0, D^0, D^1, …, D^{j-1}, we can think of this sequence as a transform of the original coefficients, known as a *wavelet transform.* Note that the total size of the transform $C^0, D^0, D^1, \ldots, D^{j-1}$ is the same as that of the original version C^j, so no extra storage is required. (However, the wavelet coefficients may require more bits to retain the accuracy of the original values.)

In general, the analysis filters A^j and B^j are not necessarily transposed multiples of the synthesis filters, as was the case for the Haar basis. Rather, A^j and B^j are formed by the matrices satisfying the relation

$$\left[\, \Phi^{j-1} \;\middle|\; \Psi^{j-1} \,\right] \left[\frac{A^j}{B^j} \right] = \Phi^j. \tag{9}$$

Note that $\left[\, P^j \;\middle|\; Q^j \,\right]$ and $\left[\frac{A^j}{B^j} \right]$ are both square matrices. Thus, combining Equations (3) and (9) gives

$$\left[\frac{A^j}{B^j} \right] = \left[\, P^j \;\middle|\; Q^j \,\right]^{-1} \tag{10}$$

Although we have not yet gotten specific about how to choose matrices A^j, B^j, P^j, and Q^j, it should be clear from Equation (10) that the two matrices in that equation must at least be invertible.

2.3 Designing a multiresolution analysis

The multiresolution analysis framework presented above is very general. In practice you often have the freedom to design a multiresolution analysis specifically suited to a particular application. The steps involved are

1. *Select the scaling functions* $\Phi^j(x)$ *for each* $j = 0, 1, \ldots$.
   This choice determines the nested approximation spaces V^j, the synthesis filters P^j, and the smoothness—that is, the number of continuous derivatives—of the analysis.
2. *Select an inner product defined on the functions in* $V^0, V^1, \ldots$.
   This choice determines the L^2 norm and the orthogonal complement spaces W^j. Although the standard inner product is the common choice, in general the inner product should be chosen to capture a measure of error that is meaningful in the context of the application.
3. *Select a set of wavelets* $\Psi^j(x)$ *that span* W^j *for each* $j = 0, 1, \ldots$.
   This choice determines the synthesis filters Q^j. Together, the synthesis filters P^j and Q^j determine the analysis filters A^j and B^j by Equation (10).

It is generally desirable to construct the wavelets to form an orthogonal basis for W^j and to have small support (the support of a function $f(x)$ is the set of points x where $f(x) \neq 0$). However, orthogonality often comes at the expense of increased supports, so a tradeoff must be made. In the case of the spline wavelets presented in the next section, the wavelets are constructed to have minimal support, but they are not orthogonal to one another (except for the piecewise-constant case). Wavelets that are both locally supported and mutually orthogonal (other than Haar wavelets) were thought to be impossible until Daubechies' ground-breaking work showing that certain families of spaces V^j actually do admit mutually orthogonal wavelets of small support [5].

3 Spline wavelets

Until now, the only specific wavelet basis we have considered is the Haar basis. Haar basis functions have a number of advantages, including

- simplicity,
- orthogonality,
- very small supports,
- nonoverlapping scaling functions (at a given level), and
- nonoverlapping wavelets (at a given level),

which make them useful in many applications. However, despite these advantages, the Haar basis is a poor choice for applications such as curve editing [8] and animation [9] because of its lack of continuity.

There are a variety of ways to construct wavelets with k continuous derivatives. One such class of wavelets can be constructed from piecewise-polynomial splines. These *spline wavelets* have been developed to a large extent by Chui and colleagues [3, 4]. The Haar basis is in fact the simplest instance of spline wavelets, resulting when the polynomial degree is set to zero.

In the following, we briefly sketch the ideas behind the construction of endpoint-interpolating B-spline wavelets. Finkelstein and Salesin [8] developed a collection of wavelets for the cubic case, and Chui and Quak [4] presented constructions for arbitrary degree. Although the derivations for arbitrary degree are too involved to present here, we give the synthesis filters for the piecewise-constant (Haar), linear, quadratic, and cubic cases in Appendix A. The next three sections parallel the three steps described in Section 2.3 for designing a multiresolution analysis.

3.1 B-spline scaling functions

Our first step is to define the scaling functions for a nested set of function spaces. We'll start with the general definition of B-splines, then specify how to make uniformly spaced, endpoint-interpolating B-splines from these. (More detailed derivations of these and other splines appear in a number of standard texts [1, 7].)

Given positive integers d and k, with $k \geq d$, and a collection of non-decreasing values $x_0, \ldots, x_{k+d+1}$ called *knots,* the *nonuniform B-spline* basis functions of degree d are defined recursively as follows. For $i = 0, \ldots, k$, and for $r = 1, \ldots, d$, let

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

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

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

The *endpoint-interpolating B-splines* of degree d on $[0, 1]$ result when the first and last $d + 1$ knots are set to 0 and 1, respectively. In

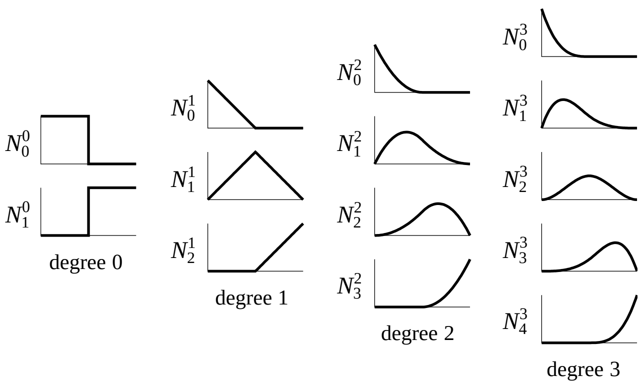

Figure 2 B-spline scaling functions for $V^1(d)$ with degree $d = 0, 1, 2,$ and 3.

this case, the functions $N_0^d(x), \ldots, N_k^d(x)$ form a basis for the space of piecewise-polynomials of degree d with $d-1$ continuous derivatives and breakpoints at the *interior knots* $x_{d+1} < x_{d+2} < \cdots < x_k$.

To make uniformly spaced B-splines, we choose $k = 2^j + d - 1$ and $x_{d+1}, \ldots, x_k$ to produce 2^j equally spaced interior intervals. This construction gives $2^j + d$ B-spline basis functions for a particular degree d and level j. We will use these functions as the endpoint-interpolating B-spline scaling functions. Figure 2 shows examples of these functions at level $j = 1$ (two interior intervals) for various degrees d. Note that the basis functions defined here are not normalized in the L^2 norm.

If $V^j(d)$ denotes the space spanned by the B-spline scaling functions of degree d with 2^j uniform intervals, it is not difficult to show that the spaces $V^0(d), V^1(d), \ldots$ are nested as required by multiresolution analysis.

The rich theory of B-splines can be used to develop expressions for the entries of the refinement matrix P^j (see Chui and Quak [4] or Quak and Weyrich [13] for details). The columns of P^j are sparse, reflecting the fact that the B-spline basis functions are locally supported. The first and last d columns of P^j are relatively complicated, but the remaining (interior) columns are shifted versions of column $d + 1$. Moreover, the entries of these interior columns are, up to a common factor of $1/2^d$, given by binomial coefficients.

Example: In the case of cubic splines ($d = 3$), the matrix P^j for $j \geq 3$ has the form

$$P^j = \frac{1}{8}\begin{bmatrix}
8 & & & & & & & & & \\
4 & 4 & & & & & & & & \\
 & 6 & 2 & & & & & & & \\
 & \frac{3}{2} & \frac{11}{2} & 1 & & & & & & \\
 & & 4 & 4 & & & & & & \\
 & & 1 & 6 & 1 & & & & & \\
 & & & 4 & 4 & & & & & \\
 & & & 1 & 6 & & & & & \\
 & & & & 4 & \cdot & & & & \\
 & & & & 1 & \cdot & 1 & & & \\
 & & & & & \cdot & 4 & & & \\
 & & & & & & 6 & 1 & & \\
 & & & & & & 4 & 4 & & \\
 & & & & & & 1 & \frac{11}{2} & \frac{3}{2} & \\
 & & & & & & & 2 & 6 & \\
 & & & & & & & & 4 & 4 \\
 & & & & & & & & & 8
\end{bmatrix}$$

where blank entries are taken to be zero, and the dots indicate that the previous column is repeated, shifted down by two rows each time.

3.2 Inner product

The second step of designing a multiresolution analysis is the choice of an inner product. We'll simply use the standard inner product here:

$$\langle f \mid g \rangle \;:=\; \int_0^1 f(x)\, g(x)\, dx.$$

3.3 B-spline wavelets

To complete our development of a multiresolution analysis based on B-splines, we need to find basis functions for the spaces W^j that are orthogonal complements to the spaces V^j. As shown in Section 2.1, the wavelets are determined by matrices Q^j satisfying Equation (5), which we repeat here for convenience:

$$[\langle \Phi^{j-1} \mid \Phi^j \rangle]\, Q^j \;=\; \mathbf{0}. \tag{11}$$

Since this is a homogeneous system of linear equations, there is not a unique solution. We must therefore impose additional conditions. To get wavelets with small supports, for example, we require each column of Q^j to have a minimal number of consecutive nonzeros. This constraint imposes a banded structure on Q^j similar to that of P^j. For each column q of Q^j, Equation (11) leads to a small homogeneous system that we solve for the non-zero entries in q. The matrices that result and the corresponding B-spline wavelets are shown in Appendix A

Finkelstein and Salesin [8] took this approach to construct cubic B-spline wavelets. Chui and Quak [4] derived slightly different spline wavelets using derivative and interpolation properties of B-splines. Note that both approaches result in semi-orthogonal wavelet bases: The wavelets are orthogonal to scaling functions at the same level, but not to each other, except in the piecewise-constant case.

3.4 B-spline filter bank

At this point, we have completed the steps in designing a multiresolution analysis. However, to use spline wavelets, we must implement a filter bank procedure incorporating the analysis filters A^j and B^j. These matrices allow us to determine C^{j-1} and D^{j-1} from C^j using matrix multiplication as in Equations (6) and (7). As discussed earlier in Section 2, the analysis filters are uniquely determined by the inverse relation in Equation (10):

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

However, as Quak and Weyrich [13] point out, when implementing the filter bank procedure for spline wavelets, it is generally *not* a good idea to form the filters A^j and B^j explicitly. Although P^j and Q^j are sparse, having only $O(d)$ entries per column, A^j and B^j are in general dense, so matrix–vector multiplication would require quadratic instead of linear time.

Fortunately, there is a better approach. The idea is to compute C^{j-1} and D^{j-1} from C^j by solving the sparse linear system

$$\left[\, P^j \mid Q^j \,\right] \left[\frac{C^{j-1}}{D^{j-1}}\right] \;=\; C^j.$$

In order to solve this system for $\left[\frac{C^{j-1}}{D^{j-1}}\right]$, we first make the matrix $\left[\, P^j \mid Q^j \,\right]$ into a banded matrix simply by interspersing the columns of P^j and Q^j. The resulting banded system can then be

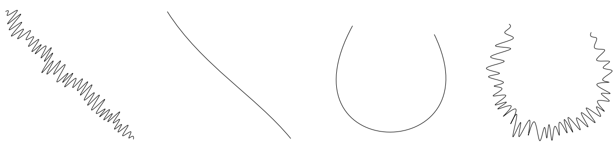

Figure 3 Changing a curve's overall sweep without affecting its character. Given the original curve (a), the system extracts the overall sweep (b). If the user modifies the sweep (c), the system can reapply the detail (d).

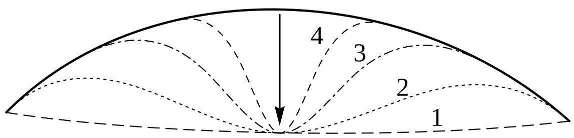

Figure 4 The middle of the dark curve is pulled, using editing at various levels of smoothing j. A change in a control point in C^1 has a very broad effect, while a change in a control point in C^4 has a narrow effect.

solved in linear time using LU decomposition [12]. Thus we can compute the entire filter bank operation without ever forming and using A^j or B^j explicitly.

4 Application III: Multiresolution curves and surfaces

We presented two applications of wavelets in Part 1: compression of one-dimensional signals and compression of two-dimensional images. Our third application of wavelets in computer graphics is curve design and editing, as described in detail by Finkelstein and Salesin [8]. Their *multiresolution curves* are built from a wavelet basis for endpoint-interpolating cubic B-splines, which we discussed in the previous section.

Multiresolution curves conveniently support a variety of operations:

- changing a curve's overall "sweep" while maintaining its fine details, or "character" (Figures 3 and 4);
- changing a curve's "character" without affecting its overall "sweep" (Figure 5);
- editing a curve at any continuous level of detail, allowing an arbitrary portion of the curve to be affected through direct manipulation;
- smoothing at continuous levels to remove undesirable features from a curve;
- approximating or "fitting" a curve within a guaranteed maximum error tolerance, for scan conversion and other applications.

Here we'll describe briefly just the first two of these operations, which fall out quite naturally from the multiresolution representation.

4.1 Editing the sweep of the curve

Editing the sweep of a curve at an integer level of the wavelet transform is simple. Let C^J be the control points of the original curve $f^J(t)$, let C^j be a low-resolution version of C^J, and let $\hat{C}^j$ be an edited version of C^j, given by $\hat{C}^j = C^j + \Delta C^j$. The edited version of the highest resolution curve $\hat{C}^J = C^J + \Delta C^J$ can be computed through synthesis:

$$\begin{aligned}\hat{C}^J &= C^J + \Delta C^J \\ &= C^J + P^J P^{J-1} \cdots P^{j+1} \Delta C^j.\end{aligned}$$

Note that editing the sweep of the curve at lower levels of smoothing j affects larger portions of the high-resolution curve $f^J(t)$. At the lowest level, when $j = 0$, the entire curve is affected. At the highest level, when $j = J$, only the narrow portion influenced by one original control point is affected. The kind of flexibility that this multiresolution editing allows is suggested in Figures 3 and 4.

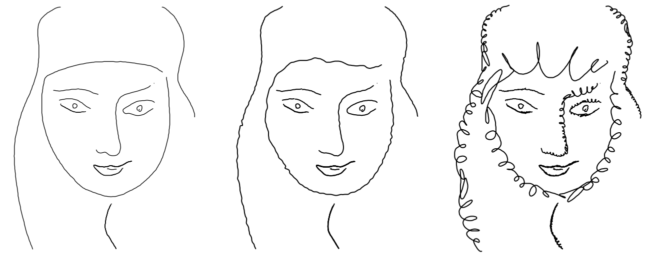

Figure 5 Changing the character of a curve without affecting its sweep.

4.2 Editing the character of the curve

Multiresolution curves also naturally support changes in the character of a curve, without affecting its overall sweep. Let C^J be the control points of a curve, and let $C^0, D^0, \ldots, D^{J-1}$ denote its wavelet transform. Editing the character of the curve is simply a matter of replacing the existing set of detail coefficients $D^j, \ldots, D^{J-1}$ with some new set $\hat{D}^j, \ldots, \hat{D}^{J-1}$, and reconstructing. To avoid coordinate-system artifacts, all detail coefficients are expressed in terms of the curve's local tangent and normal, rather than the x and y directions.

Figure 5 demonstrates how the character of curves in an illustration can be modified with various detail styles. (The interactive illustration system used to create this figure was described by Salisbury *et al.* [14].)

4.3 Multiresolution surfaces

Multiresolution editing can be extended to surfaces by using tensor products of B-spline scaling functions and wavelets. Either the standard construction or the nonstandard construction described in Part 1 for Haar basis functions can be used to form a two-dimensional basis from a one-dimensional B-spline basis. We can then edit surfaces using the same operations described for curves. For example, Figure 6 shows a bicubic tensor-product B-spline surface after altering its sweep at different levels of detail.

We can further generalize multiresolution analysis to surfaces of arbitrary topology by defining wavelets based on *subdivision surfaces*, as described by Lounsbery *et al.* [10]. Their nonorthogonal wavelet basis, in combination with the work of Eck *et al.* [6], allows any polyhedral object to be decomposed into scaling function and wavelet coefficients. Then a compression scheme similar to the one presented for images in Section 3.3 of Part 1 can be used to display the object at various levels of detail simply by leaving out small wavelet coefficients during reconstruction. An example of this technique is shown in Figure 7.

5 Conclusion

Our primer has only touched on a few of the many uses for wavelets in computer graphics. We hope this introduction to the topic has ex-

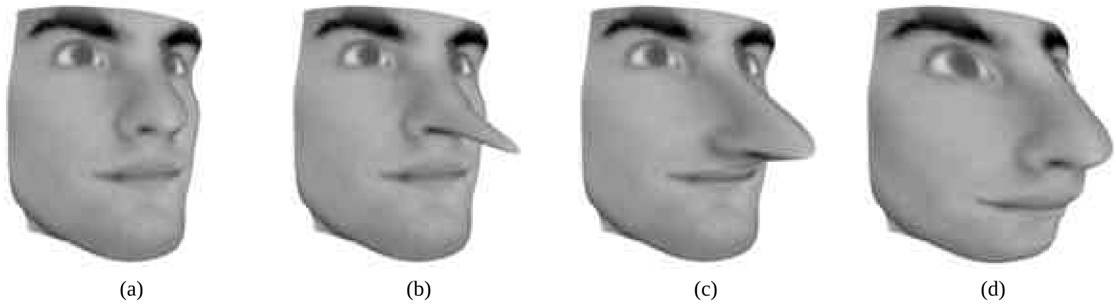

Figure 6 Surface manipulation at different levels of detail: The original surface (a) is changed at a narrow scale (b), an intermediate scale (c), and a broad scale (d).

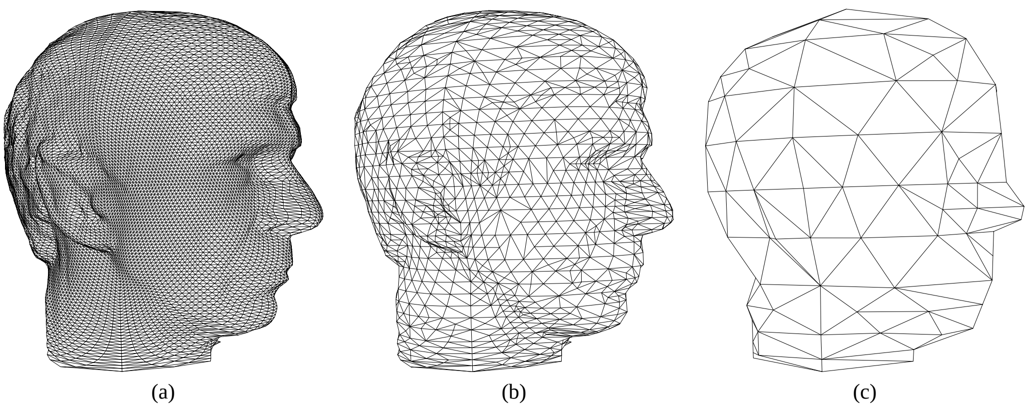

Figure 7 Surface approximation using subdivision surface wavelets: (a) the original surface, (b) an intermediate approximation, and (c) a coarse approximation.

plained enough of the fundamentals for interested readers to explore both the construction of wavelets and their application to problems in graphics and beyond. We present a more thorough discussion in a forthcoming monograph [15].

Acknowledgments

We wish to thank Adam Finkelstein, Michael Lounsbery, and Sean Anderson for help with several of the figures in this paper. Thanks also go to Ronen Barzel, Steven Gortler, Michael Shantzis, and the anonymous reviewers for their many helpful comments. This work was supported by NSF Presidential and National Young Investigator awards (CCR-8957323 and CCR-9357790), by NSF grant CDA-9123308, by an NSF Graduate Research Fellowship, by the University of Washington Royalty Research Fund (65-9731), and by industrial gifts from Adobe, Aldus, Microsoft, and Xerox.

References

[1] R. Bartels, J. Beatty, and B. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, San Francisco, 1987.

[2] Charles K. Chui. An overview of wavelets. In Charles K. Chui, editor, *Approximation Theory and Functional Analysis*, pages 47–71. Academic Press, Boston, 1991.

[3] Charles K. Chui. *An Introduction to Wavelets*. Academic Press, Boston, 1992.

[4] Charles K. Chui and Ewald Quak. Wavelets on a bounded interval. In D. Braess and L. L. Schumaker, editors, *Numerical Methods in Approximation Theory*, volume 9, pages 53–75. Birkhauser Verlag, Basel, 1992.

[5] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, October 1988.

[6] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of SIGGRAPH 95*, pages 173–182. ACM, New York, 1995.

[7] Gerald Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, Boston, third edition, 1993.

[8] Adam Finkelstein and David H. Salesin. Multiresolution curves. In *Proceedings of SIGGRAPH 94*, pages 261–268. ACM, New York, 1994.

[9] Zicheng Liu, Steven J. Gortler, and Michael F. Cohen. Hierarchical spacetime control. In *Proceedings of SIGGRAPH 94*, pages 35–42. ACM, New York, 1994.

[10] Michael Lounsbery, Tony DeRose, and Joe Warren. Multiresolution surfaces of arbitrary topological type. *ACM Transactions on Graphics*, 1996 (to appear).

[11] Stephane Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.

[12] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Fetterling. *Numerical Recipes*. Cambridge University Press, New York, second edition, 1992.

[13] Ewald Quak and Norman Weyrich. Decomposition and reconstruction algorithms for spline wavelets on a bounded interval. *Applied and Computational Harmonic Analysis*, 1(3):217–231, June 1994.

[14] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel, and David H. Salesin. Interactive pen and ink illustration. In *Proceedings of SIGGRAPH 94*, pages 101–108. ACM, New York, 1994.

[15] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann, San Francisco, 1996 (to appear).

A Details on endpoint-interpolating B-spline wavelets

This appendix presents the matrices required to apply endpoint-interpolating B-spline wavelets of low degree. (The Matlab code used to generate these matrices is available from the authors upon request.) These concrete examples should serve to elucidate the ideas presented in Section 3. To emphasize the sparse structure of the matrices, zeros have been omitted. Diagonal dots indicate that the previous column is to be repeated the appropriate number of times, shifted down by two rows for each column. The P matrices have entries relating the unnormalized scaling functions defined in Section 3, while the Q matrices have entries defining normalized, minimally supported wavelets. Columns of the Q matrices that are not represented exactly with integers are given to six decimal places.

A.1 Haar wavelets

The B-spline wavelet basis of degree 0 is simply the Haar basis described in Section 2 of Part 1. Some examples of the Haar basis scaling functions and wavelets are depicted in Figure 8. The synthesis matrices P^j and Q^j are

$$P^j = \begin{bmatrix} 1 & & & \\ 1 & & & \\ & 1 & & \\ & 1 & \ddots & \\ & & \ddots & 1 \\ & & & 1 \end{bmatrix} \qquad Q^j = \sqrt{\frac{2^j}{2}} \begin{bmatrix} 1 & & & \\ -1 & & & \\ & 1 & & \\ & -1 & \ddots & \\ & & \ddots & 1 \\ & & & -1 \end{bmatrix}$$

A.2 Endpoint-interpolating linear B-spline wavelets

Figure 9 shows a few typical scaling functions and wavelets for linear B-splines. The synthesis matrices P^j and Q^j for endpoint-interpolating linear B-spline wavelets are

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

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

$$P^{j\geq 3} = \frac{1}{2} \begin{bmatrix} 2 & & & & & \\ 1 & 1 & & & & \\ & 2 & & & & \\ & 1 & 1 & & & \\ & & 2 & & & \\ & & 1 & \ddots & & \\ & & & \ddots & & \\ & & & \ddots & 1 & \\ & & & & 2 & \\ & & & & 1 & 1 \\ & & & & & 2 \end{bmatrix}$$

$$Q^1 = \sqrt{3} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} \qquad Q^2 = \sqrt{\frac{3}{64}} \begin{bmatrix} -12 & \\ 11 & 1 \\ -6 & -6 \\ 1 & 11 \\ & -12 \end{bmatrix}$$

$$Q^{j\geq 3} = \sqrt{\frac{2^j}{72}} \begin{bmatrix} -11.022704 & & & & & \\ 10.104145 & 1 & & & & \\ -5.511352 & -6 & & & & \\ 0.918559 & 10 & 1 & & & \\ & -6 & -6 & & & \\ & 1 & 10 & & & \\ & & -6 & \ddots & & \\ & & 1 & \ddots & 1 & \\ & & & \ddots & -6 & \\ & & & & 10 & 0.918559 \\ & & & & -6 & -5.511352 \\ & & & & 1 & 10.104145 \\ & & & & & -11.022704 \end{bmatrix}$$

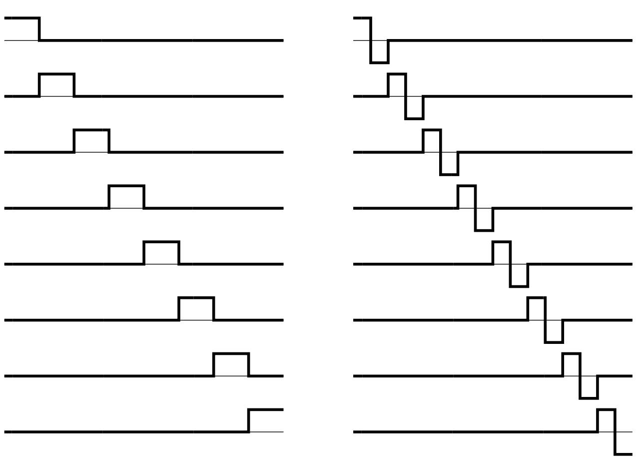

Figure 8 Piecewise-constant B-spline scaling functions and wavelets for $j = 3$.

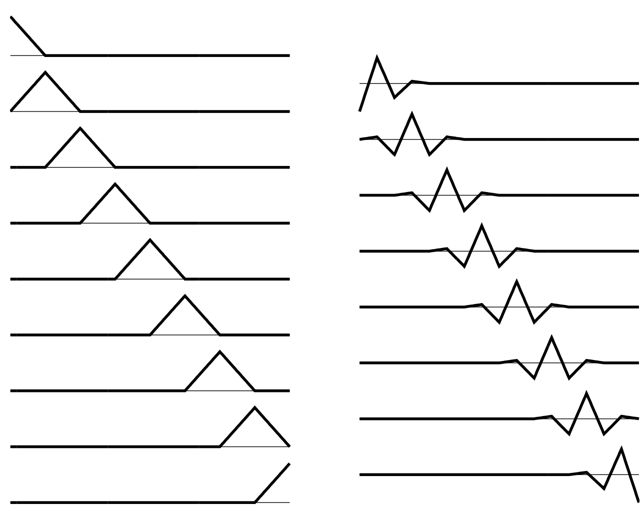

Figure 9 Linear B-spline scaling functions and wavelets for $j = 3$.

A.3 Endpoint-interpolating quadratic B-spline wavelets

Figure 10 shows some quadratic B-spline scaling functions and wavelets. The synthesis matrices P^j and Q^j for the quadratic case are

$$P^1 = \frac{1}{2}\begin{bmatrix} 2 & & \\ 1 & 1 & \\ & 1 & 1 \\ & & 2 \end{bmatrix} \qquad P^2 = \frac{1}{4}\begin{bmatrix} 4 & & & \\ 2 & 2 & & \\ & 3 & 1 & \\ & 1 & 3 & \\ & & 2 & 2 \\ & & & 4 \end{bmatrix} \qquad P^{j\geq 3} = \frac{1}{4}\begin{bmatrix} 4 & & & & & & & \\ 2 & 2 & & & & & & \\ & 3 & 1 & & & & & \\ & 1 & 3 & & & & & \\ & & 3 & 1 & & & & \\ & & 1 & 3 & & & & \\ & & & 3 & \cdot & & & \\ & & & 1 & \cdot & 1 & & \\ & & & & \cdot & 3 & & \\ & & & & & 3 & 1 & \\ & & & & & 1 & 3 & \\ & & & & & & 2 & 2 \\ & & & & & & & 4 \end{bmatrix}$$

$$Q^1 = \sqrt{\frac{5}{4}}\begin{bmatrix} -2 \\ 3 \\ -3 \\ 2 \end{bmatrix} \qquad Q^2 = \sqrt{\frac{3}{4936}}\begin{bmatrix} -144 & \\ 177 & 21 \\ -109 & -53 \\ 53 & 109 \\ -21 & -177 \\ & 144 \end{bmatrix} \qquad Q^3 = \sqrt{\frac{1}{713568}}\begin{bmatrix} -4283.828550 & & & \\ 5208.746077 & 780 & & \\ -3099.909150 & -1949 & -11 & \\ 1300.002166 & 3481 & 319 & \\ -253.384964 & -3362 & -1618 & -8.737413 \\ 8.737413 & 1618 & 3362 & 253.384964 \\ & -319 & -3481 & -1300.002166 \\ & 11 & 1949 & 3099.909150 \\ & & -780 & -5208.746077 \\ & & & 4283.828550 \end{bmatrix}$$

$$Q^{j\geq 4} = \sqrt{\frac{3 \cdot 2^j}{136088}}\begin{bmatrix} -381.872771 & & & & & & & \\ 464.322574 & 69.531439 & & & & & & \\ -276.334798 & -173.739454 & -1 & & & & & \\ 115.885924 & 310.306330 & 29 & & & & & \\ -22.587463 & -299.698329 & -147 & -1 & & & & \\ 0.778878 & 144.233164 & 303 & 29 & & & & \\ & -28.436576 & -303 & -147 & & & & \\ & 0.980572 & 147 & 303 & & -1 & & \\ & & -29 & -303 & \cdot & 29 & & \\ & & 1 & 147 & \cdot & -147 & -0.980572 & \\ & & & -29 & \cdot & 303 & 28.436576 & \\ & & & 1 & & -303 & -144.233164 & -0.778878 \\ & & & & & 147 & 299.698329 & 22.587463 \\ & & & & & -29 & -310.306330 & -115.885924 \\ & & & & & 1 & 173.739454 & 276.334798 \\ & & & & & & -69.531439 & -464.322574 \\ & & & & & & & 381.872771 \end{bmatrix}$$

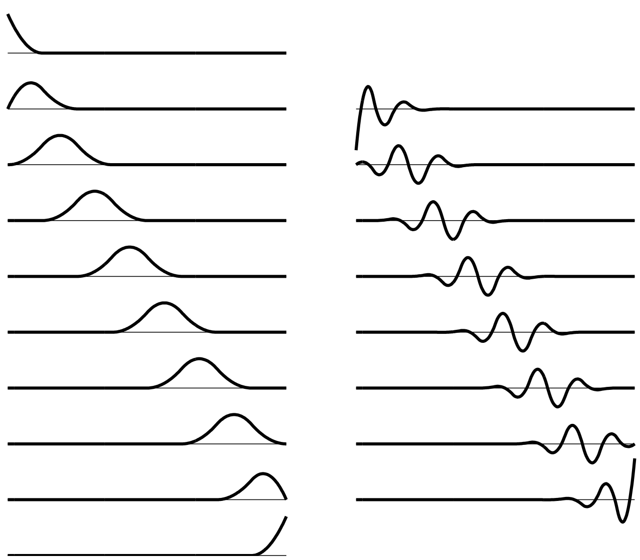

Figure 10 Quadratic B-spline scaling functions and wavelets for $j = 3$.

A.4 Endpoint-interpolating cubic B-spline wavelets

Some examples of cubic B-spline scaling functions and wavelets are shown in Figure 11. The synthesis matrices P^j and Q^j for endpoint-interpolating cubic B-spline wavelets are

$$P^1 = \frac{1}{2}\begin{bmatrix} 2 & & & \\ 1 & 1 & & \\ & 1 & 1 & \\ & & 1 & 1 \\ & & & 2 \end{bmatrix} \qquad P^2 = \frac{1}{16}\begin{bmatrix} 16 & & & & \\ 8 & 8 & & & \\ & 12 & 4 & & \\ & 3 & 10 & 3 & \\ & & 4 & 12 & \\ & & & 8 & 8 \\ & & & & 16 \end{bmatrix} \qquad P^{j\geq 3} = \frac{1}{16}\begin{bmatrix} 16 & & & & & & & & & \\ 8 & 8 & & & & & & & & \\ & 12 & 4 & & & & & & & \\ & 3 & 11 & 2 & & & & & & \\ & & 8 & 8 & & & & & & \\ & & 2 & 12 & 2 & & & & & \\ & & & 8 & 8 & & & & & \\ & & & 2 & 12 & & & & & \\ & & & & 8 & \cdot & & & & \\ & & & & 2 & \cdot & 2 & & & \\ & & & & & \cdot & 8 & & & \\ & & & & & & 12 & 2 & & \\ & & & & & & 8 & 8 & & \\ & & & & & & 2 & 11 & 3 & \\ & & & & & & & 4 & 12 & \\ & & & & & & & & 8 & 8 \\ & & & & & & & & & 16 \end{bmatrix}$$

$$Q^1 = \sqrt{7}\begin{bmatrix} 1 \\ -2 \\ 3 \\ -2 \\ 1 \end{bmatrix} \qquad Q^2 = \sqrt{\frac{315}{31196288}}\begin{bmatrix} 1368 & \\ -2064 & -240 \\ 1793 & 691 \\ -1053 & -1053 \\ 691 & 1793 \\ -240 & -2064 \\ & 1368 \end{bmatrix} \qquad Q^3 = \begin{bmatrix} 6.311454 & & & \\ -9.189342 & -1.543996 & & \\ 7.334627 & 4.226722 & 0.087556 & \\ -3.514553 & -5.585477 & -0.473604 & -0.000155 \\ 1.271268 & 6.059557 & 1.903267 & 0.019190 \\ -0.259914 & -4.367454 & -4.367454 & -0.259914 \\ 0.019190 & 1.903267 & 6.059557 & 1.271268 \\ -0.000155 & -0.473604 & -5.585477 & -3.514553 \\ & 0.087556 & 4.226722 & 7.334627 \\ & & -1.543996 & -9.189342 \\ & & & 6.311454 \end{bmatrix}$$

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

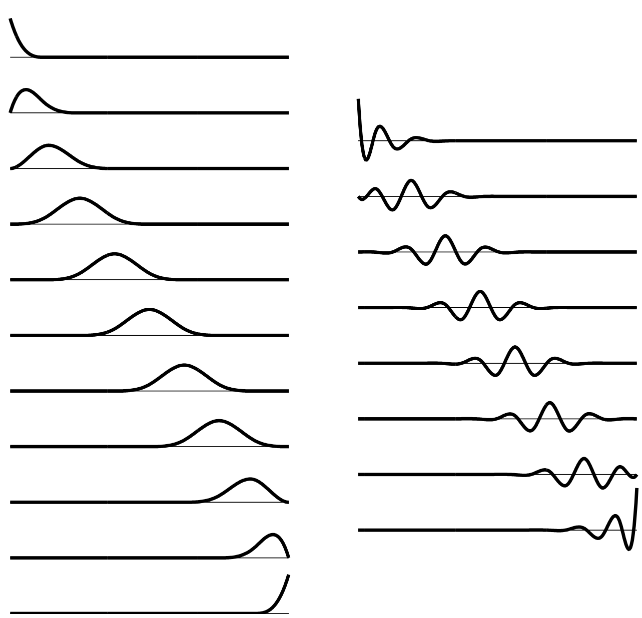

Figure 11 Cubic B-spline scaling functions and wavelets for $j = 3$.