Wavelets for Computer Graphics: A Primer

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

*Department of Applied Mathematics
†Department of Computer Science and Engineering
University of Washington
Seattle, Washington 98195

1 Introduction

Wavelets are a mathematical tool for hierarchically decomposing functions. Wavelets allow any 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.

Although wavelets have their roots in approximation theory [6] and signal processing [15], they have recently been applied to many problems in computer graphics. These graphics applications include image editing [1] and compression [7], automatic level-of-detail control for editing and rendering curves and surfaces [9, 11, 14], surface reconstruction from contours [16], and fast methods for solving simulation problems in global illumination [3, 12, 19] and animation [13]. This primer is intended to provide those working in computer graphics with some intuition for what wavelets are, as well as to present the mathematical foundations necessary for studying and using them.

The remainder of the primer is laid out as follows. In Section 2, we set the stage by presenting the simplest form of wavelets, the Haar basis. We cover both one- and two-dimensional wavelet transforms and basis functions. Then in Section 3, we discuss image compression as a first application of wavelets. In Section 4, we present the mathematical theory of wavelets based on Mallat's "multiresolution analysis" [15], using the Haar wavelets as a running example. The original formulation of multiresolution analysis is applicable to functions defined on the infinite real line. Following Lounsbery *et al.* [14], we generalize the theory to functions defined on a bounded interval by making use of block matrices instead of convolution filters. Section 5 introduces smooth wavelets based on bounded-interval B-splines. We make use of these spline wavelets in a second application of wavelets, which we present in Section 6: multiresolution editing of curves and surfaces. Finally, we include as appendices a brief review of linear algebra and a summary of the matrices required for B-spline wavelets of low degree.

2 The Haar wavelet basis

The Haar basis is the simplest wavelet basis. We will first discuss how a one-dimensional function can be decomposed using Haar wavelets, and then describe the actual basis functions. After that, we will extend both the decomposition and the basis functions to two dimensions, so that we can make use of wavelets for image compression in Section 3.

2.1 The one-dimensional Haar wavelet transform

To get a sense for how wavelets work, let's start out with a simple example. Suppose we are given a one-dimensional "image" with a resolution of 4 pixels, having the following pixel values:

$$\begin{bmatrix} 8 & 4 & 1 & 3 \end{bmatrix}$$

This image can be represented in the *Haar basis*, the simplest wavelet basis, as follows. Start by averaging the pixels together, pairwise, to get the new lower-resolution image with pixel values:

$$\begin{bmatrix} 6 & 2 \end{bmatrix}$$

Clearly, some information has been lost in this averaging and downsampling process. In order to be able to recover the original four pixel values from the two averaged pixels, we need to store some *detail coefficients*, which capture that missing information. In our example, we will choose 2 for the first detail coefficient, since the average we computed is 2 less than 8 and 2 more than 4. This single number allows us to recover the first two pixels of our original 4-pixel image. Similarly, the second detail coefficient is -1, since $2 + (-1) = 1$ and $2 - (-1) = 3$.

Summarizing, we have so far decomposed the original image into a lower-resolution 2-pixel image version and detail coefficients as follows:

Resolution	Averages	Detail coefficients
4	$\begin{bmatrix} 8 & 4 & 1 & 3 \end{bmatrix}$	
2	$\begin{bmatrix} 6 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & -1 \end{bmatrix}$

Repeating this process recursively on the averages gives the full decomposition:

Resolution	Averages	Detail coefficients
4	$\begin{bmatrix} 8 & 4 & 1 & 3 \end{bmatrix}$	
2	$\begin{bmatrix} 6 & 2 \end{bmatrix}$	$\begin{bmatrix} 2 & -1 \end{bmatrix}$
1	$\begin{bmatrix} 4 \end{bmatrix}$	$\begin{bmatrix} 2 \end{bmatrix}$

Finally, we will define the *wavelet transform* of the original 4-pixel image to be the single coefficient representing the overall average of the original image, followed by the detail coefficients in order of increasing resolution. Thus, for the one-dimensional Haar basis, the wavelet transform of our original 4-pixel image is given by

$$\begin{bmatrix} 4 & 2 & 2 & -1 \end{bmatrix}.$$

Note that no information has been gained or lost by this process: The original image had 4 coefficients, and so does the transform. Also note that, given the transform, we can reconstruct the image to any resolution by recursively adding and subtracting the detail coefficients from the lower-resolution versions.

Storing the wavelet transform of the image, rather than the image itself, has a number of advantages. One advantage of storing the wavelet transform of the image is that often a large number of the detail coefficients turn out to be very small in magnitude, as in the larger example of Figure 1. Truncating, or removing, these small coefficients from the representation introduces only small errors in the reconstructed image, giving a form of "lossy" image compression. We will discuss this particular application of wavelets in Section 3, once we have presented the one- and two-dimensional Haar basis functions.

2.2 One-dimensional Haar wavelet basis functions

In the previous section we treated one-dimensional images as sequences of coefficients. Alternatively, we can think of images as piecewise-constant functions on the half-open interval $[0, 1)$. In order to do so, we will use the concept of a *vector space* from linear algebra (see Appendix A for a refresher on linear algebra). A one-pixel image is just a function that is constant over the entire interval $[0, 1)$; we'll let V^0 be the space of all these functions. A two-pixel image has two constant pieces over the intervals $[0, 1/2)$ and $[1/2, 1)$. We'll call the space containing all these functions V^1. If we continue in this manner, the space V^j will include all piecewise-constant functions on the interval $[0, 1)$, with the interval divided equally into 2^j different pieces.

We can now think of every one-dimensional image with 2^j pixels as being an element, or vector, in V^j. Note that because these vectors are all functions defined on the unit interval, every vector in V^j is also contained in V^{j+1}. For example, we can always

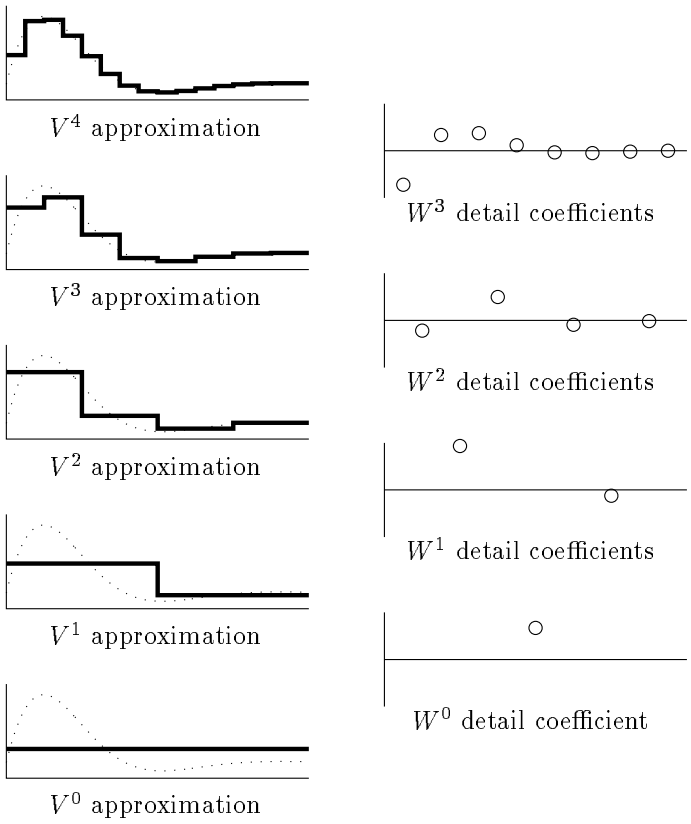


Figure 1 A sequence of decreasing-resolution approximations to a function (left), along with the detail coefficients required to recapture the finest approximation (right). Note that in regions where the true function is close to being flat, a piecewise-constant approximation is a good one, and so the corresponding detail coefficients are relatively small.

describe a piecewise-constant function with two intervals as a piecewise-constant function with four intervals, with each interval in the first function corresponding to a pair of intervals in the second. Thus, the spaces V^j are nested; that is,

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

This nested set of spaces V^j is called a *multiresolution analysis*.

Now we need to define a basis for each vector space V^j. The basis functions for the spaces V^j are called *scaling functions*, and are usually denoted by the symbol ϕ. A simple basis for V^j is given by 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

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

As an example, the four box functions forming the basis for V^2 are shown in Figure 2.

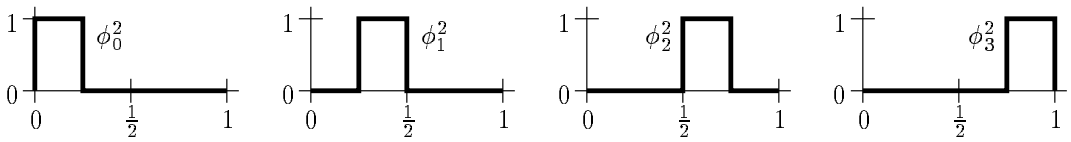


Figure 2 The box basis for V^2.

The next step is to choose an inner product defined on the vector spaces V^j. The standard inner product,

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

for two elements $f, g \in V^j$ will do quite well for our running example. We can now define a new vector space W^j as the *orthogonal complement* of V^j in V^{j+1}. In other words, we will let W^j be the space of all functions in V^{j+1} that are orthogonal to all functions in V^j under the chosen inner product (see Appendix A for more on inner products and orthogonality).

A collection of functions $\psi_i^j(x)$ spanning W^j are called *wavelets*. These basis functions have two important properties:

- the basis functions ψ_i^j of W^j, together with the basis functions ϕ_i^j of V^j, form a basis for V^{j+1}; and
- every basis function ψ_i^j of W^j is orthogonal to every basis function ϕ_i^j of V^j under the chosen inner product.[1]

[1] This property is convenient, but not strictly necessary for most of our development of wavelets.

Informally, we can think of the wavelets in W^j as a means for representing the parts of a function in V^{j+1} that cannot be represented in V^j. Thus, the "detail coefficients" of Section 2.1 are really coefficients of the wavelet basis functions.

The wavelets corresponding to the box basis are known as the *Haar wavelets*, given by

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

where

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

Figure 3 shows the two Haar wavelets spanning W^1.

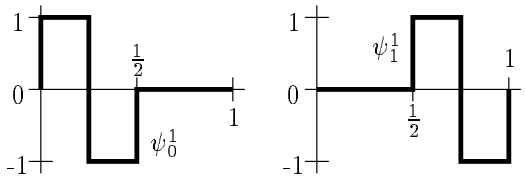


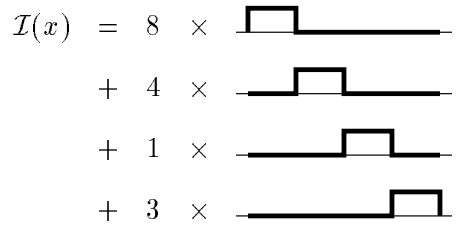
Figure 3 The Haar wavelets for W^1.

Before going on, let's run through our example from Section 2.1 again, but now applying these more sophisticated ideas.

We begin by expressing our original image $\mathcal{I}(x)$ as a linear combination of the box basis functions in V^2:

$$\mathcal{I}(x) \;=\; c_0^2\,\phi_0^2(x) \;+\; c_1^2\,\phi_1^2(x) \;+\; c_2^2\,\phi_2^2(x) \;+\; c_3^2\,\phi_3^2(x).$$

A more graphical representation is



Note that the coefficients $c_0^2, \ldots, c_3^2$ are just the four original pixel values $[8, 4, 1, 3]$.

We can rewrite the expression for $\mathcal{I}(x)$ in terms of basis functions in V^1 and W^1,

using pairwise averaging and differencing:

$$\mathcal{I}(x) = c_0^1\,\phi_0^1(x) + c_1^1\,\phi_1^1(x) + d_0^1\,\psi_0^1(x) + d_1^1\,\psi_1^1(x)$$

$$= 6 \times$$

$$+ 2 \times$$

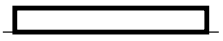
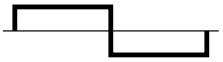
$$+ 2 \times$$

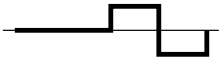
$$+ -1 \times$$


These four coefficients should look familiar as well.

Finally, we'll rewrite $\mathcal{I}(x)$ as a sum of basis functions in V^0, W^0, and W^1:

$$\mathcal{I}(x) = c_0^0\,\phi_0^0(x) + d_0^0\,\psi_0^0(x) + d_0^1\,\psi_0^1(x) + d_1^1\,\psi_1^1(x)$$

$$= 4 \times$$

$$+ 2 \times$$

$$+ 2 \times$$

$$+ -1 \times$$


Once again, these four coefficients are the Haar wavelet transform of the original image. The four functions shown above constitute the Haar basis for V^2. Instead of using the usual four box functions, we can use these four functions representing the overall average, the broad detail, and the two types of finer detail possible in a function in V^2. The Haar basis for V^j with $j > 2$ includes these functions as well as narrower translates of the wavelet $\psi(x)$.

Note that for the sake of clarity, we have not normalized the basis functions given above (see Appendix A for an explanation of normalization). If we normalize the basis functions, we replace our earlier definitions with

$$\begin{aligned} \phi_i^j(x) &:= 2^{j/2}\,\phi(2^j x - i) \\ \psi_i^j(x) &:= 2^{j/2}\,\psi(2^j x - i). \end{aligned}$$

This normalization also modifies the wavelet transform somewhat: When considering two neighboring values, rather than dividing their sum by 2 and their difference by 2, we divide both the sum and the difference by $\sqrt{2}$.

2.3 Two-dimensional Haar wavelet transforms

In preparation for image compression, we need to generalize Haar wavelets to two dimensions. First, we will consider how to perform a wavelet transform of the pixel values in a two-dimensional image. Then in the next section we will describe the scaling functions and wavelets that form a two-dimensional wavelet basis.

There are two ways we can use wavelets to transform or decompose the pixel values within an image. Each of these is a generalization to two dimensions of the one-dimensional wavelet transform described in Section 2.1. Note that a multi-dimensional wavelet transform is frequently referred to in the literature as a *wavelet decomposition*.

To obtain the *standard decomposition* [2] of an image, we first apply the one-dimensional wavelet transform to each row of pixel values. This operation gives us an average value along with detail coefficients for each row. Next, we treat these transformed rows as if they were themselves an image, and apply the one-dimensional transform to each column. The resulting values are all detail coefficients except for a single overall average coefficient. We illustrate each step of the standard decomposition in Figure 4.

The second type of two-dimensional wavelet transform, called the *non-standard decomposition*, alternates between operations on rows and columns. First, we perform one step of horizontal pairwise averaging and differencing on the pixel values in each row of the image. Next, we apply vertical pairwise averaging and differencing to each column of the result. To complete the transformation, we repeat this process recursively on the quadrant containing averages in both directions. Figure 5 shows all the steps involved in the non-standard decomposition of an image.

2.4 Two-dimensional Haar basis functions

The two methods of decomposing a two-dimensional image yield coefficients that correspond to two different sets of basis functions. The standard decomposition of an image gives coefficients for a basis formed by the *standard construction* [2] of a two-dimensional basis. Similarly, the non-standard decomposition gives coefficients for the *non-standard construction* of basis functions.

The standard construction of a two-dimensional wavelet basis consists of all possible tensor products of one-dimensional basis functions. For example, when we start with the one-dimensional Haar basis for V^2, we get the two-dimensional basis for V^2 that is shown in Figure 6.

The *non-standard construction* of a two-dimensional basis proceeds by first defining

Figure 4 Standard decomposition of an image.

Figure 5 Non-standard decomposition of an image.

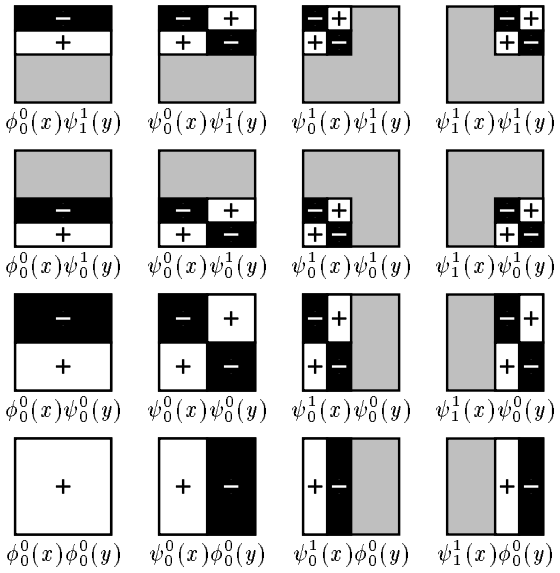


Figure 6 The standard construction of a two-dimensional Haar wavelet basis for V^2. In the unnormalized case, functions are $+1$ where plus signs appear, -1 where minus signs appear, and 0 in gray regions.

a two-dimensional scaling function,

$$\phi\phi(x,y) \;:=\; \phi(x)\,\phi(y),$$

and three wavelet functions,

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

The basis consists of a single coarse scaling function along with all possible scales and translates of the three wavelet functions. This construction results in the basis for V^2 shown in Figure 7.

We have presented both the standard and non-standard approaches to wavelet transforms and basis functions because they each have advantages. The standard decomposition of an image is appealing because it can be accomplished simply by performing one-dimensional transforms on all the rows and then on all the columns. On the other hand, it is slightly more efficient to compute the non-standard decomposition of an image. Each step of the non-standard decomposition computes one quarter of the coefficients that the previous step did, as opposed to one half in the standard case.

Another consideration is the *support* of each basis function, meaning the portion of each function's domain where that function is non-zero. All of the non-standard

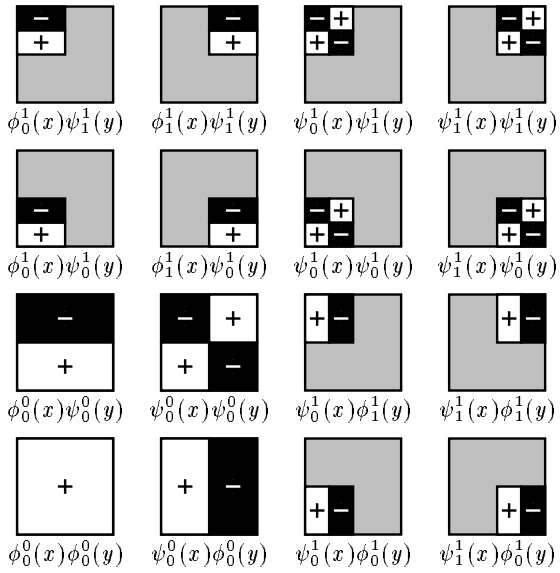


Figure 7 The non-standard construction of a two-dimensional Haar wavelet basis for V^2.

basis functions have square supports, while some of the standard basis functions have non-square supports. Depending upon the application, one of these choices may be more favorable than another.

3 Application I: Image compression

In this section we discuss image compression as an application of wavelets to computer graphics. First we define what we mean by compression, and then we outline a general method and apply it to images using a two-dimensional Haar wavelet basis. The approach presented here is only introductory; for a more complete treatment of wavelet image compression, see the article by DeVore *et al.* [7].

3.1 Compression

The goal of compression is to express an initial set of data using some smaller set of data, either with or without a loss of information. For instance, suppose we are given a function $f(x)$ expressed as a weighted sum of basis functions $u_0(x), \ldots, u_{M-1}(x)$:

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

The data set in this case consists of the coefficients $c_0, \ldots, c_{M-1}$. We would like to find a function approximating $f(x)$ but requiring fewer coefficients, perhaps by using a different basis. That is, given a user-specified error tolerance ϵ (for lossless

compression, $\epsilon = 0$), we are looking for

$$\widetilde{f}(x) \quad = \quad \sum_{i=0}^{M'-1} \widetilde{c}_i \widetilde{u}_i(x)$$

such that $M' < M$ and $||f(x) - \widetilde{f}(x)|| \leq \epsilon$ for some norm (see Appendix A for more on norms). In general, one could attempt to construct a set of basis functions $\widetilde{u}_0, \ldots, \widetilde{u}_{M-1}$ that would provide a good approximation with few coefficients. We will focus instead on the simpler problem of finding a good approximation in a fixed basis.

3.2 L_2 compression

One form of the compression problem is to order the coefficients $c_0, \ldots, c_{M-1}$ so that for every $M' < M$, the first M' elements of the sequence give the best approximation $\widetilde{f}(x)$ to $f(x)$ as measured in the L_2 norm. As we show here, the solution to this problem is straightforward if the basis is orthonormal.

Let σ be a permutation of $0, \ldots, M-1$, and let $\widetilde{f}(x)$ be a function that uses the coefficients corresponding to the first M' numbers of the permutation σ:

$$\widetilde{f}(x) \quad = \quad \sum_{i=0}^{M'-1} c_{\sigma(i)} u_{\sigma(i)}.$$

The square of the L_2 error in this approximation is given by

$$\begin{aligned}
\left|\left| f(x) - \widetilde{f}(x) \right|\right|_2^2 &= \langle f(x) - \widetilde{f}(x) \,|\, f(x) - \widetilde{f}(x) \rangle \\
&= \left\langle \sum_{i=M'}^{M-1} c_{\sigma(i)} u_{\sigma(i)} \;\middle|\; \sum_{j=M'}^{M-1} c_{\sigma(j)} u_{\sigma(j)} \right\rangle \\
&= \sum_{i=M'}^{M-1} \sum_{j=M'}^{M-1} c_{\sigma(i)} c_{\sigma(j)} \langle u_{\sigma(i)} \,|\, u_{\sigma(j)} \rangle \\
&= \sum_{i=M'}^{M-1} (c_{\sigma(i)})^2.
\end{aligned}$$

The last step follows from the assumption that the basis is orthonormal. We conclude that in order to minimize this error for a given M', the best choice for σ is the permutation that sorts the coefficients in order of decreasing magnitude; that is, σ satisfies $|c_{\sigma(0)}| \geq \cdots \geq |c_{\sigma(M-1)}|$.

Figure 1 demonstrated how a one-dimensional function could be transformed by a filter bank operation into coefficients representing the function's overall average and various resolutions of detail. Now we repeat the process, this time using normalized

Haar basis functions satisfying $\langle \phi_i^j(x) \,|\, \phi_i^j(x) \rangle = 1$ and $\langle \psi_i^j(x) \,|\, \psi_i^j(x) \rangle = 1$. We can apply L_2 compression to the resulting coefficients simply by removing or ignoring the coefficients with smallest magnitude. By varying the amount of compression, we obtain a sequence of approximations to the original function, as shown in Figure 8.

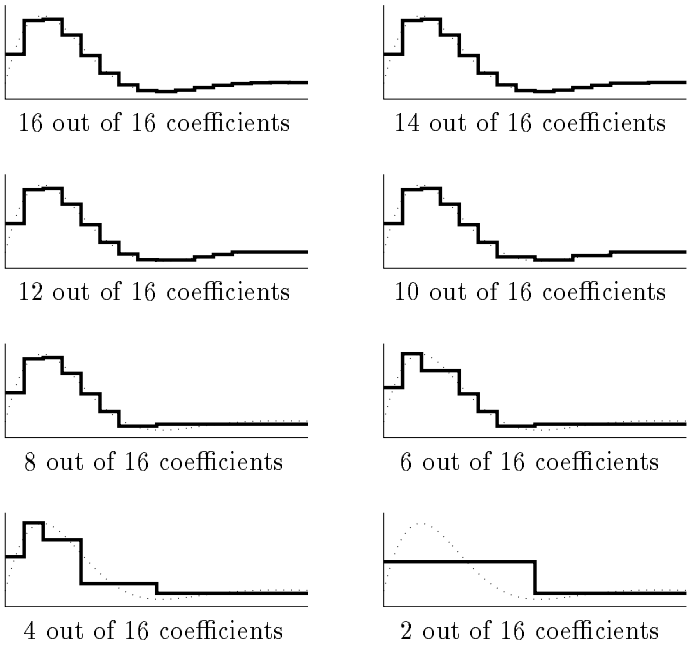


Figure 8 Coarse approximations to a function obtained using L_2 compression: detail coefficients are removed in order of increasing magnitude.

3.3 Wavelet image compression in the L_2 norm

Wavelet image compression using the L_2 norm can be summarized in three steps:

1. Compute coefficients representing an image in a normalized two-dimensional Haar basis.
2. Sort the coefficients in order of decreasing magnitude to produce the sequence $c_{\sigma(0)}, \ldots, c_{\sigma(M-1)}$.
3. Starting with $M' = M$, find the least M' for which $\sum_{i=M'}^{M-1} (c_{\sigma(i)})^2 \leq \epsilon^2$.

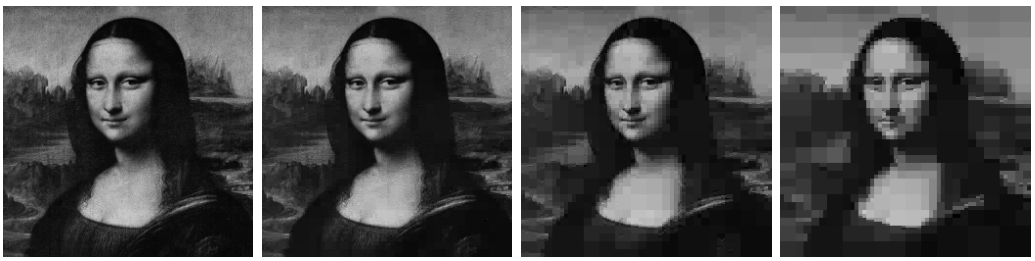
The first step is accomplished by applying either of the two-dimensional Haar wavelet transforms described in Section 2.3, making sure to use normalized basis functions. Any standard sorting technique will work for the second step; however, for large images sorting becomes exceedingly slow.

The pseudocode below outlines a more efficient method that only partitions subsets of the coefficients when necessary to achieve the desired level of error. We use μ to

denote a set of coefficients under consideration ("maybes"), and κ to denote a set of coefficients that will be used for the compressed image ("keepers"). The square of the L_2 error is accumulated in s. This method is similar to quicksort in the way it partitions coefficients using a pivot element, but the process is only repeated on one side of the partition.

$\kappa \leftarrow \emptyset$
$\mu \leftarrow \{c_0, \ldots, c_{M-1}\}$
$s \leftarrow 0$
repeat
  $\rho \leftarrow$ pivot chosen from μ
  $\alpha \leftarrow \{c \in \mu : |c| > |\rho|\}$
  $\beta \leftarrow \{c \in \mu : |c| \leq |\rho|\}$
  $\Delta s \leftarrow \sum_{c \in \beta} c^2$
  if $s + \Delta s > \epsilon^2$ **then**
    $\kappa \leftarrow \kappa \cup \alpha$
    $\mu \leftarrow \beta$
  else
    $\mu \leftarrow \alpha$
    discard coefficients in β
    $s \leftarrow s + \Delta s$
  end if
until $s \approx \epsilon^2$

Another efficient approach to L_2 compression is to repeatedly discard all coefficients smaller in magnitude than a threshold, increasing the threshold by a factor of two in each iteration, until the allowable level of error is achieved. This method was used to produce the images in Figure 9. These images demonstrate the high compression ratios wavelets offer, as well as some of the artifacts they introduce.



(a) (b) (c) (d)

Figure 9 L_2 wavelet image compression: The original image (a) can be represented using (b) 21% of its wavelet coefficients, with 5% relative L_2 error; (c) 4% of its coefficients, with 10% relative L_2 error; and (d) 1% of its coefficients, with 15% relative L_2 error.

3.4 Wavelet image compression in other L_p norms

Images that have been compressed using the L_2 norm can exhibit large errors so long as they are confined to small areas. An alternative approach to compression uses the L_∞ norm to ensure that no pixel has error greater than ϵ. Shown below is the pseudocode for a "greedy" L_∞ compression scheme:

```
for each pixel (x, y) do
  δ[x][y] ← 0
end for
for each coefficient c_i do
  δ' ← δ + |error from eliminating c_i|
  if δ' < ε everywhere then
    eliminate coefficient c_i
    δ ← δ'
  end if
end for
```

Note that this algorithm's results depend on the order in which coefficients are visited. One could imagine obtaining very different images (and amounts of compression) by eliminating fine-scale wavelet coefficients first, rather than small coefficients first, for example. One could also imagine running a more sophisticated constrained optimization procedure whose goal is to select the minimum number of coefficients subject to the error bound.

DeVore *et al.* [7] suggest that the L_1 norm is best suited to the task of image compression. Rather than repeat their results, we refer the interested reader to their article. We also note that the algorithm for L_∞ compression above can easily be modified to measure L_1 error: Instead of eliminating coefficients when $\delta' < \epsilon$ everywhere, do so only when the sum of all entries in δ' is smaller than ϵ.

4 Multiresolution analysis

The Haar wavelets we have discussed so far are just one among many sets of basis functions 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. 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 Appendix A for a brief review.

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*.

4.1 A matrix formulation for refinement

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,

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

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

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

where N is the dimension of W^j.

The condition requiring 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 constant matrix P^j such that

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

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 if V^j and V^{j-1} have dimensions M and M', respectively, then P^j is an $M \times M'$ matrix.

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 a product of the scaling functions $\Phi^j(x)$ and another constant matrix Q^j:

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

Thus, each wavelet at level $j-1$ is also expressible as a linear combination of "finer" scaling functions at level j. If V^j and V^{j-1} have dimensions M and M', respectively, then W^{j-1} has dimension $M - M'$, and Q^j must be an $M \times (M - M')$ matrix.

> **Example:** In the Haar basis, at a particular level j there are $M = 2^j$ scaling functions and $N = 2^j$ 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

$$\begin{aligned} \phi(x) &= \sum_i p_i \, \phi(2x - i) \\ \psi(x) &= \sum_i q_i \, \phi(2x - i). \end{aligned}$$

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

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

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 synthesis filters P^j, the wavelet synthesis filters Q^j are somewhat constrained. 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. We can rewrite this condition using the compact matrix notation of Appendix A to get

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

Substituting Equation (2) into Equation (4) yields

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

The columns of Q^j must therefore form a basis for the *null space* of $[\langle \Phi^{j-1} \mid \Phi^j \rangle]$ (see Golub and Van Loan [10] for a discussion of null spaces). There are a multitude of bases for the null space of a matrix, implying that there are many different wavelet bases for a given space W^j. Ordinarily, we impose further constraints in addition to the orthogonality requirement to uniquely determine the Q^j matrices. For example, the Haar wavelet synthesis matrix can be found by requiring the least number of non-zero entries in each column.

4.2 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.

Consider a function in some scaling function space V^n. 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 $C^n = [c_0^n \cdots c_{M-1}^n]^{\mathrm{T}}$. The coefficients c_i^n 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^{n-1} of C^n with a smaller number of coefficients M'. The standard approach for creating the M' values of C^{n-1} is to use some form of linear filtering and downsampling on the M entries of C^n. This process can be expressed as a matrix equation

$$C^{n-1} \quad = \quad A^n \, C^n \tag{6}$$

where A^n is a constant $M' \times M$ matrix.

Since C^{n-1} contains fewer entries than C^n, it is intuitively clear that some amount of detail is lost in this filtering process. If A^n is appropriately chosen, it is possible to capture the lost detail as another column matrix D^{n-1}, computed by

$$D^{n-1} \quad = \quad B^n \, C^n \tag{7}$$

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

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

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

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

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

$$A^n = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & & & & \\ & & 1 & 1 & & & & \\ & & & & \ddots & & & \\ & & & & & 1 & 1 & \\ & & & & & & & 1 \quad 1 \end{bmatrix}$$

$$B^n = \frac{1}{2}\begin{bmatrix} 1 & -1 & & & & & & \\ & & 1 & -1 & & & & \\ & & & & \ddots & & & \\ & & & & & 1 & -1 & \\ & & & & & & & 1 \quad -1 \end{bmatrix},$$

where blank entries are taken to be zero, and the dots indicate that the previous row is repeated, shifted right by two columns each time. In fact, $A^n = (P^n)^{\mathrm{T}}/2$ and $B^n = (Q^n)^{\mathrm{T}}/2$ for the Haar basis.

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

$$\begin{aligned} c_k^{n-1} &= \sum_\ell a_{\ell-2k} c_\ell^j \\ d_k^{n-1} &= \sum_\ell b_{\ell-2k} c_\ell^j \end{aligned}$$

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^n and B^n, respectively. Similarly, Equation (8) for reconstruction often appears as

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

Note that the procedure for splitting C^n into a low-resolution part C^{n-1} and a detail part D^{n-1} can be applied recursively to the low-resolution version C^{n-1}. Thus, the original coefficients can be expressed as a hierarchy of lower-resolution versions $C^0, \ldots, C^{n-1}$ and details $D^0, \ldots, D^{n-1}$, as shown in Figure 10. This recursive process is known as a *filter bank*.

Since the original coefficients C^n can be recovered from the sequence $C^0, D^0, D^1, \ldots, D^{n-1}$, this sequence can be thought of as a transform of the original coefficients,

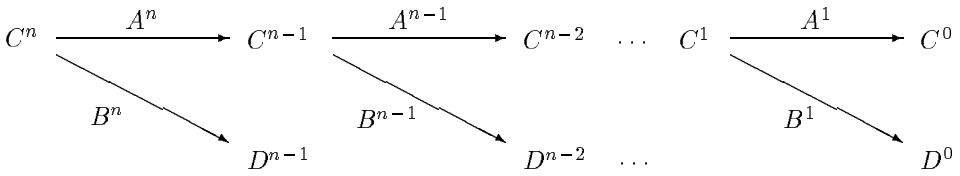


Figure 10 The filter bank.

known as a *wavelet transform*. Note that the total size of the transform C^0, D^0, ..., D^{n-1} is the same as that of the original version C^n, so no extra storage is required.[2]

In general, the analysis filters A^j and B^j are not necessarily 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 been specific about how to choose matrices A^j, B^j, P^j, and Q^j, it should be clear from Equation (10) that $\left[\frac{A^j}{B^j}\right]$ and $\left[\, P^j \;\middle|\; Q^j \,\right]$ must at least be invertible.

4.3 Designing a multiresolution analysis

The multiresolution analysis framework presented above is very general. In practice one often has 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 vector spaces V^j, the synthesis filters P^j, and the smoothness — that is, the number of continuous derivatives — of the analysis.

[2] However, the wavelet coefficients may require more bits to retain the accuracy of the original values.

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 have small support and to form an orthonormal basis for W^j. However, orthonormality often comes at the expense of increased supports, so a tradeoff must be made. In the case of the spline wavelets presented in Section 5, the wavelets are constructed to have minimal support, but they are not orthonormal (except for the piecewise-constant case). Wavelets that are both locally supported and orthonormal (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 orthonormal wavelets of small support [6].

5 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 compact supports,
- non-overlapping scaling functions (at a given level), and
- non-overlapping 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 [9] and animation [13] 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 [4, 5]. 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 [9] have developed a collection of wavelets for the cubic case, and Chui and Quak [5] present constructions for

arbitrary degree. Although the derivations for arbitrary degree are too involved to present here, in Appendix B we give the synthesis filters for the piecewise-constant (Haar), linear, quadratic, and cubic cases. The next three sections should parallel the three steps of designing a multiresolution analysis that were described in Section 4.3.

5.1 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, and then specify how to make uniformly spaced, endpoint-interpolating B-splines from these.

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 *non-uniform B-spline* basis functions of degree d are defined recursively as follows. For $i = 0, \ldots, k$, and for $r = 1, \ldots, d$, let[3]

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

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 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* $x_{d+1}, \ldots, x_k$ [8].

To make uniformly spaced B-splines that are refinable, 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 11 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.

If we denote by $V^j(d)$ 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 [5] 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.

[3] The fractions in these equations are taken to be 0 when their denominators are 0.

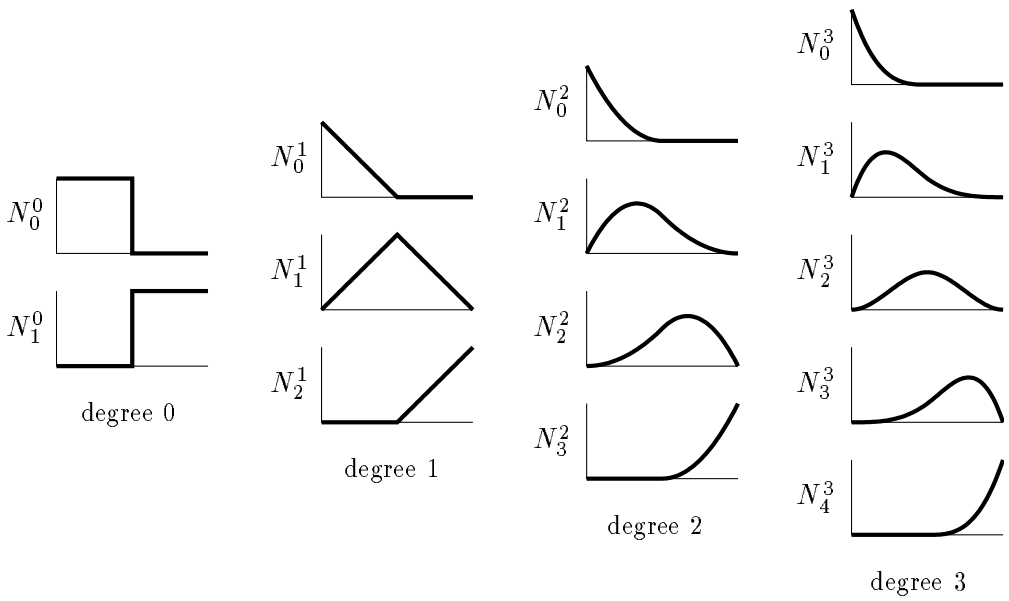


Figure 11 B-spline scaling functions for $V^1(d)$ with degree $d = 0, 1, 2$, and 3.

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

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

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.

5.2 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,

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

5.3 B-spline wavelets

To complete our development of a B-spline multiresolution analysis, we need to find basis functions for the spaces W^j that are orthogonal complements to the spaces V^j. As shown in Section 4.1, the wavelets are determined by matrices Q^j satisfying Equation (5), which we repeat here for convenience:

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

Since this is a homogeneous system of linear equations, there is not a unique solution. We must therefore impose additional conditions. We could, for example, require the columns of Q^{j+1} to be sparse, and further require a minimal number of consecutive non-zeros. This approach is taken by Finkelstein and Salesin [9] in their construction of cubic spline wavelets. This same approach was used to construct the wavelets summarized in Appendix B. The basic idea is to construct Q^{j+1} a column at a time. The matrices in Appendix B reveal a simple structure for the locations of non-zero entries; the values are determined by solving the linear system of constraints implied by Equation (11). Chui and Quak [5] use a slightly different characterization based on derivative and interpolation properties of B-splines.

5.4 B-spline filter bank

At this point, we have completed the steps in designing a multiresolution analysis. However, in order to use spline wavelets we will need to 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 in Section 4, the analysis filters are uniquely determined by the inverse relation in Equation (10):

$$\left[\begin{array}{c} A^j \\ \hline B^j \end{array}\right] \quad = \quad \left[\, P^j \;\middle|\; Q^j \,\right]^{-1}.$$

However, 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 that matrix–vector multiplication would require quadratic instead of linear time.

Fortunately, there is a better approach. The idea is to first notice that C^{j-1} and D^{j-1} can be computed from C^j by solving the sparse linear system

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

The matrix $\left[\, P^j \mid Q^j \,\right]$ can then be made into a banded matrix simply by interspersing the columns of P^j and Q^j. The resulting banded system can be solved in linear time using LU decomposition [17].

6 Application II: Multiresolution curves and surfaces

Our second application of wavelets in computer graphics is that of curve design and editing, as described in detail by Finkelstein and Salesin [9]. 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:

- the ability to change the overall "sweep" of a curve while maintaining its fine details, or "character" (Figure 12);
- the ability to change a curve's "character" without affecting its overall "sweep" (Figure 14);
- the ability to edit a curve at any continuous level of detail, allowing an arbitrary portion of the curve to be affected through direct manipulation;
- continuous levels of smoothing, in which undesirable features are removed from a curve;
- curve approximation, or "fitting," within a guaranteed maximum error tolerance, for scan conversion and other applications.

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

6.1 Editing the sweep of the curve

Editing the sweep of a curve at an integer level of the wavelet transform is simple. Let C^n be the control points of the original curve $f^n(t)$, let C^j be a low-resolution

version of C^n, and let $\widehat{C}^j$ be an edited version of C^j, given by $\widehat{C}^j = C^j + \Delta C^j$. The edited version of the highest-resolution curve $\widehat{C}^n = C^n + \Delta C^n$ can be computed through synthesis:

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

Note that editing the sweep of the curve at lower levels of smoothing j affects larger portions of the high-resolution curve $f^n(t)$. At the lowest level, when $j = 0$, the entire curve is affected; at the highest level, when $j = n$, 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 12 and 13.

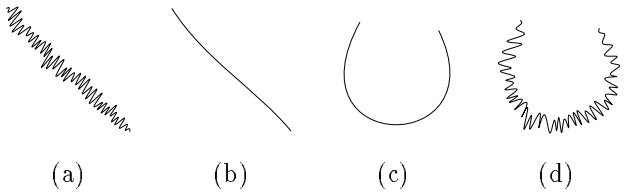


Figure 12 Changing the overall sweep of a curve without affecting its character. Given the original curve (a), the system extracts the overall sweep (b). If the user modifies the sweep (c), the system can re-apply the detail (d).

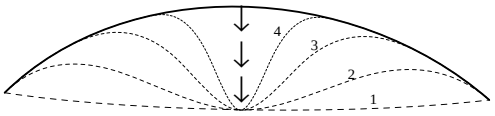


Figure 13 The middle of the dark curve is pulled, using editing at integer levels 1, 2, 3, and 4.

6.2 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^n be the control points of a curve, and let C^0, ..., C^{n-1}, D^0, ..., D^{n-1} denote the components of its multiresolution decomposition. Editing the character of the curve is simply a matter of replacing the existing set of detail functions $D^j, \ldots, D^{n-1}$ with some new set $\widehat{D}^j, \ldots, \widehat{D}^{n-1}$.

Figure 14 demonstrates how the character of curves in an illustration can be modified with the same (or different) detail styles. (The interactive illustration system used to create this figure is described by Salisbury *et al.* [18].)

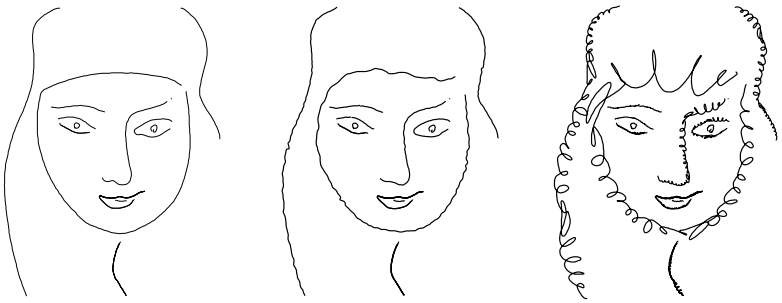
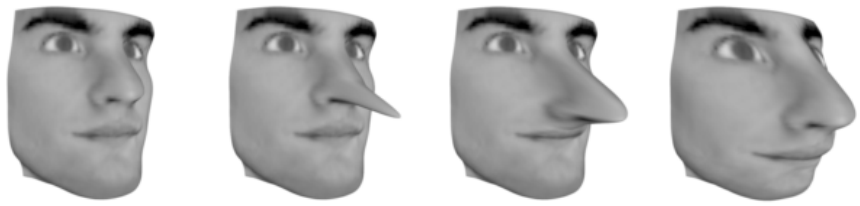


Figure 14 Changing the character of a curve without affecting its sweep.

6.3 Multiresolution surfaces

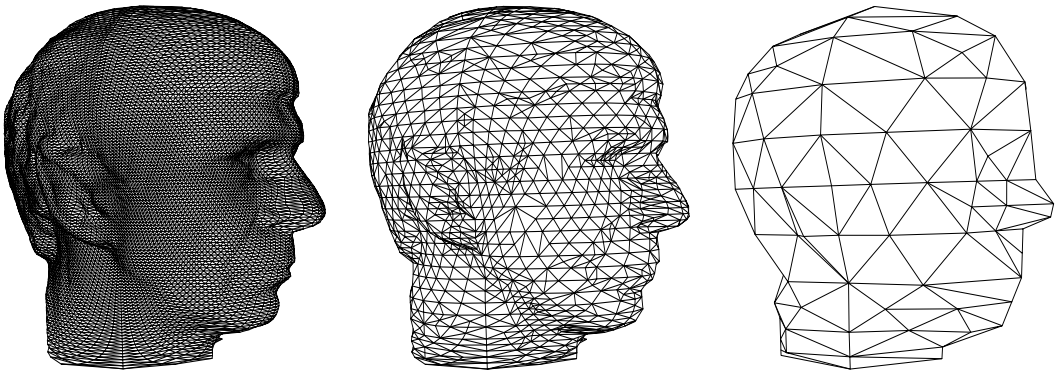
Multiresolution editing can be extended to surfaces by using tensor products of B-spline scaling functions and wavelets. Either the standard construction or the non-standard construction from Section 2.4 can be used to form a two-dimensional basis from a one-dimensional B-spline basis. We can then edit surfaces using the same operations that were described for curves in the previous sections. For example, in Figure 15 a bicubic tensor-product B-spline surface is shown after altering its sweep at different levels of detail.



(a) (b) (c) (d)

Figure 15 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).

Multiresolution analysis can be further generalized to surfaces of arbitrary topology by defining wavelets on *subdivision surfaces*, as described by Lounsbery *et al.* [14]. This method allows any polyhedral object to be decomposed into scaling function and wavelet coefficients. Then a compression scheme similar to that presented for images in Section 3 can be used to display the object at various levels of detail simply by leaving out small wavelet coefficients. An example of this technique is shown in Figure 16.



(a) (b) (c)

Figure 16 Surface approximation using subdivision surface wavelets: The original surface (a), an intermediate approximation (b), and a coarse approximation (c).

Acknowledgements

We wish to thank Adam Finkelstein, Michael Lounsbery, and Sean Anderson for help with several of the figures in this paper. This work was supported by NSF Presidential and National Young Investigator awards (CCR-8957323 and CCR-9357790), by an NSF Graduate Research Fellowship, by the University of Washington Graduate Research and Royalty Research Funds (75-1721 and 65-9731), and by industrial gifts from Adobe, Aldus, and Xerox.

References

[1] Deborah Berman, Jason Bartell, and David Salesin. Multiresolution painting and compositing. In *SIGGRAPH 94 Conference Proceedings*, Computer Graphics Annual Conference Series, pages 85–90, July 1994.

[2] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Communications on Pure and Applied Mathematics*, 44:141–183, 1991.

[3] Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose. Wavelet radiance. In *Proceedings of the Fifth Eurographics Workshop on Rendering*, pages 287–302, Darmstadt, Germany, June 1994.

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

[5] 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, Basel, 1992. Birkhauser Verlag.

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

[7] R. DeVore, B. Jawerth, and B. Lucier. Image compression through wavelet transform coding. *IEEE Transactions on Information Theory*, 38(2):719–746, March 1992.

[8] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, Boston, third edition, 1992.

[9] Adam Finkelstein and David H. Salesin. Multiresolution curves. In *SIGGRAPH 94 Conference Proceedings*, Computer Graphics Annual Conference Series, pages 261–268, July 1994.

[10] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, second edition, 1989.

[11] Steven J. Gortler and Michael F. Cohen. Variational modeling with wavelets. Technical Report CS-TR-94-456, Princeton University, Department of Computer Science, 1994.

[12] Steven J. Gortler, Peter Schröder, Michael F. Cohen, and Pat Hanrahan. Wavelet radiosity. In *SIGGRAPH 93 Conference Proceedings*, Computer Graphics Annual Conference Series, pages 221–230, August 1993.

[13] Zicheng Liu, Steven J. Gortler, and Michael F. Cohen. Hierarchical spacetime control. In *SIGGRAPH 94 Conference Proceedings*, Computer Graphics Annual Conference Series, pages 35–42, July 1994.

[14] Michael Lounsbery, Tony DeRose, and Joe Warren. Multiresolution surfaces of arbitrary topological type. Technical Report 93-10-05B, University of Washington, Department of Computer Science and Engineering, October 1993.

[15] 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.

[16] David Meyers. Multiresolution tiling. In *Proceedings of Graphics Interface '94*, pages 25–32, Banff, Alberta, May 1994.

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

[18] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel, and David H. Salesin. Interactive pen and ink illustration. In *SIGGRAPH 94 Conference Proceedings*, Computer Graphics Annual Conference Series, pages 101–108, July 1994.

[19] Peter Schröder, Steven J. Gortler, Michael F. Cohen, and Pat Hanrahan. Wavelet projections for radiosity. In *Proceedings of the Fourth Eurographics Workshop on Rendering*, pages 105–114, Paris, France, 1993.

A Linear algebra review

Multiresolution analysis relies heavily on fundamental ideas from linear algebra. We've included this appendix to remind you of a few important facts.

A.1 Vector spaces

The starting point for linear algebra is the notion of a *vector space*, which can be loosely defined as follows: A collection V of things is a vector space (over the reals) if:

1. For all $a, b \in \mathbb{R}$ and for all $u, v \in V$, $au + bv \in V$.
2. There exists a unique element $\mathbf{0} \in V$ such that
   - for all $u \in V$, $0u = \mathbf{0}$; and
   - for all $u \in V$, $\mathbf{0} + u = u$.
3. (Remaining axioms omitted; most of these are necessary to guarantee that multiplication and addition behave as expected.)

The elements of a vector space V are called *vectors*, and the element $\mathbf{0}$ is called the zero vector. The vectors may be geometric vectors, or they may be functions, as is the case when discussing wavelets and multiresolution analysis.

A.2 Bases and dimension

A collection of vectors $u_1, u_2, \ldots$ in a vector space V are said to be *linearly independent* if

$$c_1 u_1 + c_2 u_2 + \cdots = \mathbf{0} \quad \Longleftrightarrow \quad c_1 = c_2 = \cdots = 0.$$

A collection $u_1, u_2, \ldots \in V$ of linearly independent vectors is a *basis* for V if every $v \in V$ can be written as

$$v \;=\; \sum_i c_i u_i$$

for some real numbers $c_1, c_2, \ldots$. Intuitively speaking, linear independence means that the vectors are not redundant, and a basis consists of a minimal set of complete vectors.

If a basis for V has a finite number of elements $u_1, \ldots, u_m$, then V is *finite-dimensional* and its dimension is m. Otherwise, V is said to be *infinite-dimensional*.

> **Example:** $\mathbb{R}^3$ is a 3-dimensional space, and $e_1 = (1, 0, 0)$, $e_2 = (0, 1, 0)$, $e_3 = (0, 0, 1)$ is a basis for it.

Example: The set of all functions continuous on $[0,1]$ is an infinite-dimensional vector space. We'll call this space $C[0,1]$.

A.3 Inner products and orthogonality

When dealing with geometric vectors from the vector space $\mathbb{R}^3$, the "dot product" operation has a number of uses. The generalization of the dot product to arbitrary vector spaces is called an *inner product*. Formally, an inner product $\langle \cdot \,|\, \cdot \rangle$ on a vector space V is any map from $V \times V$ to $\mathbb{R}$ that is:

- symmetric: $\langle u \,|\, v \rangle = \langle v \,|\, u \rangle$
- bilinear: $\langle au + bv \,|\, w \rangle = a\langle u \,|\, w \rangle \; + \; b\langle v \,|\, w \rangle$
- positive definite: $\langle u \,|\, u \rangle > 0$ for all $u \neq \mathbf{0}$.

A vector space together with an inner product is called, not surprisingly, an *inner product space*.

Example: It is straightforward to show that the dot product on $\mathbb{R}^3$ defined by

$$\langle (a_1, a_2, a_3) \,|\, (b_1, b_2, b_3) \rangle \quad := \quad a_1 b_1 + a_2 b_2 + a_3 b_3 \tag{12}$$

satisfies the requirements of an inner product.

Example: The following "standard" inner product on $C[0,1]$ plays a central role in most formulations of multiresolution analysis:

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

Example: To further emphasize how general an inner product can be, here is a rather bizarre one on the space of twice-differentiable functions on $[0,1]$:

$$\langle f \,|\, g \rangle \quad := \quad \int_0^1 \frac{d^2 f(x)}{dx^2} \, \frac{d^2 g(x)}{dx^2} \, dx.$$

One of the most important uses of the inner product is to formalize the idea of orthogonality: Two vectors u, v in an inner product space are said to be *orthogonal* if $\langle u \,|\, v \rangle = 0$. It is not difficult to show that a collection $u_1, u_2, \ldots$ of mutually orthogonal vectors must be linearly independent, suggesting that orthogonality is a strong form of linear independence. An *orthogonal basis* is one consisting of mutually orthogonal vectors.

A.4 Norms and normalization

A *norm* is a function that measures the length of vectors. In a finite-dimensional vector space, we typically use the norm $||u|| := \langle u \mid u \rangle^{1/2} = \sqrt{u \cdot u}$. If we are working with a function space such as $C[0,1]$, we ordinarily use one of the L_p norms, defined as

$$||u||_p \quad := \quad \left(\int_0^1 |u(x)|^p \, dx \right)^{1/p}.$$

In the limit as p tends to infinity, we get what is known as the *max-norm*:

$$||u||_\infty \quad := \quad \max_{x \in [0,1]} u(x).$$

Even more frequently used is the L_2 norm, which can also be written as $||u||_2 = \langle u \mid u \rangle^{1/2}$ if we are using the standard inner product.

A vector u with $||u|| = 1$ is said to be *normalized*. If we have an orthogonal basis composed of vectors that are normalized in the L_2 norm, the basis is called *orthonormal*. Stated concisely, a basis $u_1, u_2, \ldots$ is orthonormal if

$$\langle u_i \mid u_j \rangle \quad = \quad \delta_{ij},$$

where δ_{ij} is called the Kronecker delta, and is defined to be 1 if $i = j$, and 0 otherwise.

> **Example:** The vectors $e_1 = (1,0,0)$, $e_2 = (0,1,0)$, $e_3 = (0,0,1)$ form an orthonormal basis for the inner product space $\mathbb{R}^3$ endowed with the dot product of Equation (12).

A.5 Duals

With every basis $u_1, u_2, \ldots$ for an inner product space V, there is a unique associated basis $\overline{u}_1, \overline{u}_2, \ldots$ called the *dual basis*, characterized by the relation

$$\langle u_i \mid \overline{u}_j \rangle \quad = \quad \delta_{ij}.$$

Dual bases are the key to solving the following important problem: Given an arbitrary vector v, find the coefficients $c_1, c_2, \ldots$ such that

$$v \quad = \quad \sum_i c_i u_i. \tag{13}$$

To compute the j-th coefficient c_j, simply take the inner product of both sides of Equation (13) with $\overline{u}_j$:

$$\begin{aligned}
\langle v \,|\, \overline{u}_j \rangle &= \left\langle \sum_i c_i u_i \;\middle|\; \overline{u}_j \right\rangle \\
&= \sum_i c_i \langle u_i \,|\, \overline{u}_j \rangle \\
&= \sum_i c_i \delta_{ij} \\
&= c_j.
\end{aligned}$$

In short, the coefficients in the expansion of v are given by $c_j = \langle v \,|\, \overline{u}_j \rangle$.

Orthonormal bases are so useful because they are *self-dual*; that is, they satisfy $\overline{u}_i = u_i$ for all i.

A.6 Computing the duals

To see how to compute the duals, let's define some new notation. Let $A = [a_1 \;\; a_2 \;\; \cdots]$ and $B = [b_1 \;\; b_2 \;\; \cdots]$ be two row matrices whose entries are vectors in an inner product space, and define $[\langle A \,|\, B \rangle]$ as the matrix whose ij-th entry is $\langle a_i \,|\, b_j \rangle$.

> **Example:** If $A(x) = [a_1(x) \;\; a_2(x) \;\; \cdots]$ and $B(x) = [b_1(x) \;\; b_2(x) \;\; \cdots]$ have entries in the vector space $C[0,1]$ endowed with the standard inner product, then the ij-th entry of $[\langle A \,|\, B \rangle]$ is
>
> $$[\langle A \,|\, B \rangle]_{ij} = \int_0^1 a_i(x)\, b_j(x)\, dx.$$

Armed with this notation, if we gather the elements $u_1, u_2, \ldots$ of a basis together into a row matrix $U = [u_1 \;\; u_2 \;\; \cdots]$, it is straightforward to verify that the dual basis $\overline{U} = [\overline{u}_1 \;\; \overline{u}_2 \;\; \cdots]$ can be computed from

$$\overline{U} = U\, [\langle U \,|\, U \rangle]^{-1}.$$

B Details on endpoint-interpolating B-spline wavelets

This appendix presents the matrices required to make use of endpoint-interpolating B-spline wavelets of low degree.[4] These concrete examples should serve to elucidate the ideas presented in Section 5. In order 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. All of the matrix entries given below are exact, except for those in the last two Q matrices in the cubic case. Finally, note that the matrices given here correspond to unnormalized basis functions. The P matrices have entries relating the scaling functions defined in Section 5, while the entries of the Q matrices are scaled to be integers.

B.1 Haar wavelets

The B-spline wavelet basis of degree 0 is simply the Haar basis described in Sections 2 and 4. Some examples of the Haar basis scaling functions and wavelets are depicted in Figure 17. The synthesis matrices P^j and Q^j are given below.

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

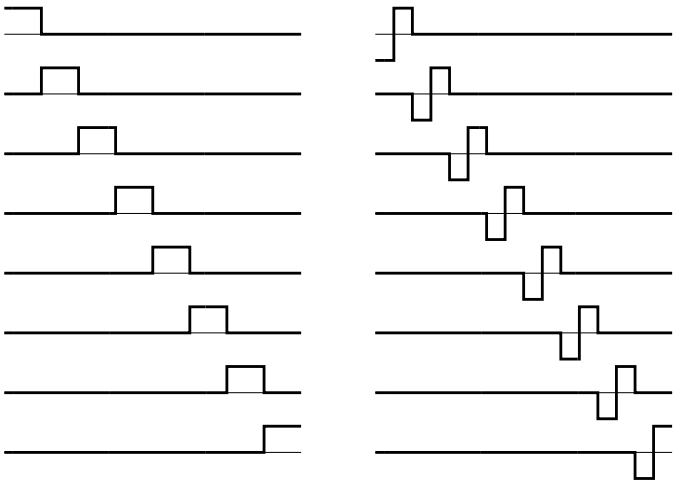


Figure 17 The piecewise-constant B-spline (Haar) scaling functions and wavelets for $j = 3$.

[4] The Matlab code used to generate these matrices is available from the authors upon request.

B.2 Endpoint-interpolating linear B-spline wavelets

Figure 18 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 given below.

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

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

$$P^{j\geq 3} = \frac{1}{2}\begin{bmatrix} 2 & & & & & & \\ 1 & 1 & & & & & \\ & 2 & & & & & \\ & 1 & 1 & & & & \\ & & 2 & & & & \\ & & 1 & \cdot & & & \\ & & & \cdot & & & \\ & & & & \cdot & 1 & \\ & & & & & 2 & \\ & & & & & 1 & 1 \\ & & & & & & 2 \end{bmatrix} \qquad Q^{j\geq 3} = \begin{bmatrix} -12 & & & & & & & \\ 11 & 1 & & & & & & \\ -6 & -6 & & & & & & \\ 1 & 10 & 1 & & & & & \\ & -6 & -6 & & & & & \\ & 1 & 10 & & & & & \\ & & -6 & \cdot & & & & \\ & & 1 & & \cdot & 1 & & \\ & & & & \cdot & -6 & & \\ & & & & & 10 & 1 & \\ & & & & & -6 & -6 & \\ & & & & & 1 & 11 & \\ & & & & & & -12 & \end{bmatrix}$$

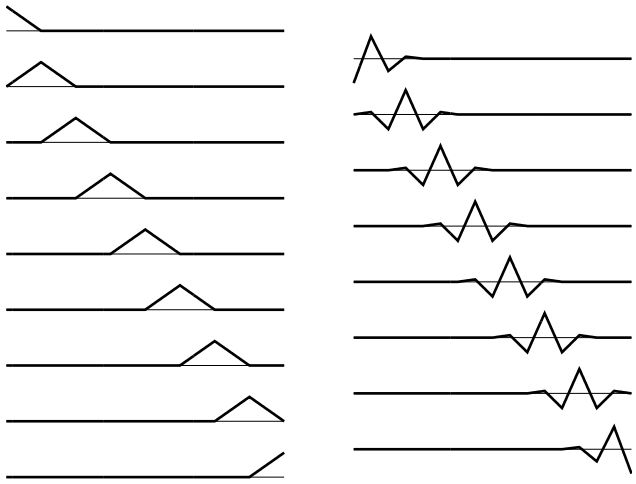


Figure 18 The linear B-spline scaling functions and wavelets for $j = 3$.

B.3 Endpoint-interpolating quadratic B-spline wavelets

Figure 19 shows some of the quadratic B-spline scaling functions and wavelets. The synthesis matrices P^j and Q^j for the quadratic case are given below.

$$P^1 = \frac{1}{2}\begin{bmatrix} 2 & & \\ 1 & 1 & \\ & 1 & 1 \\ & & 2 \end{bmatrix} \qquad Q^1 = \begin{bmatrix} 2 \\ -3 \\ 3 \\ -2 \end{bmatrix}$$

$$P^2 = \frac{1}{4}\begin{bmatrix} 4 & & & \\ 2 & 2 & & \\ & 3 & 1 & \\ & 1 & 3 & \\ & & 2 & 2 \\ & & & 4 \end{bmatrix} \qquad Q^2 = \begin{bmatrix} 144 & \\ -177 & -21 \\ 109 & 53 \\ -53 & -109 \\ 21 & 177 \\ & -144 \end{bmatrix}$$

$$Q^3 = \begin{bmatrix} -75504 & & & \\ 91806 & 10920 & & \\ -54637 & -27286 & -154 & \\ 22913 & 48734 & 4466 & \\ -4466 & -47068 & -22652 & -154 \\ 154 & 22652 & 47068 & 4466 \\ & -4466 & -48734 & -22913 \\ & 154 & 27286 & 54637 \\ & & -10920 & -91806 \\ & & & 75504 \end{bmatrix}$$

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

$$Q^{j\geq 4} = \begin{bmatrix} -75504 & & & & & & & & & \\ 91806 & 10920 & & & & & & & & \\ -54637 & -27286 & -154 & & & & & & & \\ 22913 & 48734 & 4466 & & & & & & & \\ -4466 & -47068 & -22638 & -154 & & & & & & \\ 154 & 22652 & 46662 & 4466 & & & & & & \\ & -4466 & -46662 & -22638 & & & & & & \\ & 154 & 22638 & 46662 & & -154 & & & & \\ & & -4466 & -46662 & \cdot & 4466 & & & & \\ & & 154 & 22638 & \cdot & -22638 & -154 & & & \\ & & & -4466 & \cdot & 46662 & 4466 & & & \\ & & & 154 & & -46662 & -22652 & -154 & & \\ & & & & & 22638 & 47068 & 4466 & & \\ & & & & & -4466 & -48734 & -22913 & & \\ & & & & & 154 & 27286 & 54637 & & \\ & & & & & & -10920 & -91806 & & \\ & & & & & & & 75504 & & \end{bmatrix}$$

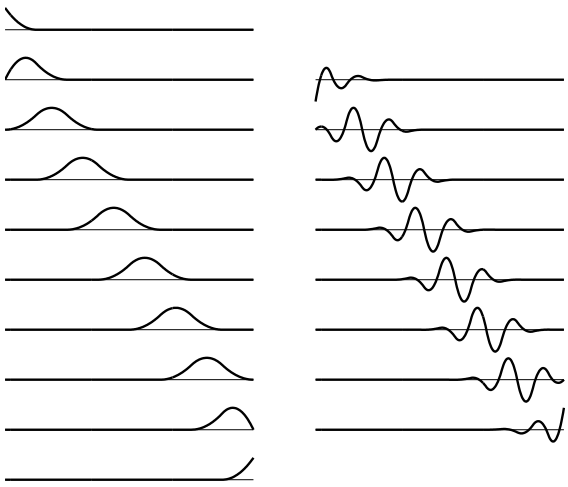


Figure 19 The quadratic B-spline scaling functions and wavelets for $j = 3$.

B.4 Endpoint-interpolating cubic B-spline wavelets

Some examples of cubic B-spline scaling functions and wavelets are shown in Figure 20. The synthesis matrices P^j and Q^j for endpoint-interpolating cubic B-spline wavelets are given below. The matrices Q^3 and $Q^{j \geq 4}$ are approximate; entries were rounded after each column was scaled to make its largest entry 1,000,000.

$$P^1 = \frac{1}{2}\begin{bmatrix} 2 & & & \\ 1 & 1 & & \\ & 1 & 1 & \\ & & 1 & 1 \\ & & & 2 \end{bmatrix} \qquad Q^1 = \begin{bmatrix} 1 \\ -2 \\ 3 \\ -2 \\ 1 \end{bmatrix}$$

$$P^2 = \frac{1}{16}\begin{bmatrix} 16 & & & & \\ 8 & 8 & & & \\ & 12 & 4 & & \\ & 3 & 10 & 3 & \\ & & 4 & 12 & \\ & & & 8 & 8 \\ & & & & 16 \end{bmatrix} \qquad Q^2 = \begin{bmatrix} -1368 & \\ 2064 & 240 \\ -1793 & -691 \\ 1053 & 1053 \\ -691 & -1793 \\ 240 & 2064 \\ & -1368 \end{bmatrix}$$

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

$$Q^3 \approx \begin{bmatrix} -686823 & & & \\ 1000000 & -254803 & & \\ -798167 & 697530 & 14449 & \\ 382460 & -921763 & -78158 & 17 \\ -138342 & 1000000 & 314093 & -2088 \\ 28284 & -720755 & -720755 & 28284 \\ -2088 & 314093 & 1000000 & -138342 \\ 17 & -78158 & -921763 & 382460 \\ & 14449 & 697530 & -798167 \\ & & -254803 & 1000000 \\ & & & -686823 \end{bmatrix}$$

$$Q^{j \geq 4} \approx \begin{bmatrix} -686823 & & & & & & & & & \\ 1000000 & 256326 & & & & & & & & \\ -798167 & -701422 & -15882 & & & & & & & \\ 382460 & 925781 & 85895 & 41 & & & & & & \\ -138342 & -1000000 & -343712 & -5110 & & & & & & \\ 28284 & 711467 & 771597 & 69115 & 41 & & & & & \\ -2088 & -297591 & -1000000 & -325750 & -5110 & & & & & \\ 17 & 62856 & 758322 & 761705 & 69115 & & & & & \\ & -4647 & -323843 & -1000000 & -325750 & & & & & \\ & 37 & 68691 & 761705 & 761705 & & 41 & & & \\ & & -5079 & -325750 & -1000000 & & -5110 & & & \\ & & 41 & 69115 & 761705 & \cdot & 69115 & 41 & & \\ & & & -5110 & -325750 & \ddots & -325750 & -5079 & & \\ & & & 41 & 69115 & \cdot & 761705 & 68691 & 37 & \\ & & & & -5110 & & -1000000 & -323843 & -4647 & \\ & & & & 41 & & 761705 & 758322 & 62856 & 17 \\ & & & & & & -325750 & -1000000 & -297591 & -2088 \\ & & & & & & 69115 & 771597 & 711467 & 28284 \\ & & & & & & -5110 & -343712 & -1000000 & -138342 \\ & & & & & & 41 & 85895 & 925781 & 382460 \\ & & & & & & & -15882 & -701422 & -798167 \\ & & & & & & & & 256326 & 1000000 \\ & & & & & & & & & -686823 \end{bmatrix}$$

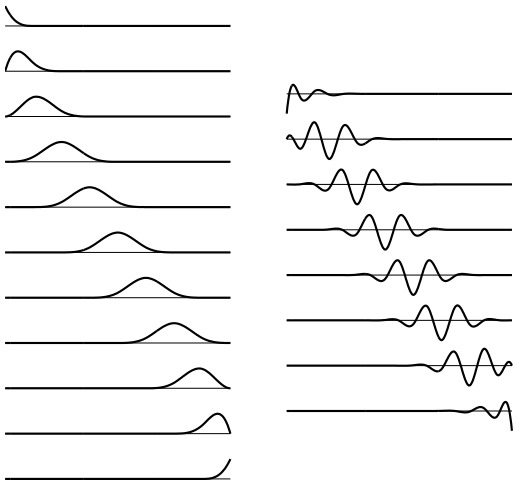


Figure 20 The cubic B-spline scaling functions and wavelets for $j = 3$.

Contents