Wavelets for Computer Graphics: A Primer
Part 1†

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 offer an elegant technique for representing the levels of detail present. This primer is intended to provide people 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. In Part 1, we discuss the simple case of Haar wavelets in one and two dimensions, and show how they can be used for image compression. In Part 2, we will present the mathematical theory of multiresolution analysis, then develop spline wavelets and describe their use in multiresolution curve and surface editing.

Although wavelets have their roots in approximation theory [5] and signal processing [13], they have recently been applied to many problems in computer graphics. These graphics applications include image editing [1], image compression [6], and image querying [10]; automatic level-of-detail control for editing and rendering curves and surfaces [7, 8, 12]; surface reconstruction from contours [14]; and fast methods for solving simulation problems in animation [11] and global illumination [3, 4, 9, 15]. For a discussion of wavelets that goes beyond the scope of this primer, we refer readers to our forthcoming monograph [16].

We set the stage here by first presenting the simplest form of wavelets, the Haar basis. We cover one-dimensional wavelet transforms and basis functions, and show how these tools can be used to compress the representation of a piecewise-constant function. Then we discuss two-dimensional generalizations of the Haar basis, and demonstrate how to apply these wavelets to image compression.

Because linear algebra is central to the mathematics of wavelets, we briefly review important concepts in Appendix A.

2 Wavelets in one dimension

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. Finally, we show how using the Haar wavelet decomposition leads to a straightforward technique for compressing a one-dimensional function.

2.1 One-dimensional Haar wavelet transform

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

$$\begin{bmatrix} 9 & 7 & 3 & 5 \end{bmatrix}$$

We can represent this image in the *Haar basis* by computing a wavelet transform. To do this, we first average the pixels together, pairwise, to get the new lower resolution image with pixel values

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

Clearly, some information has been lost in this averaging process. To recover the original four pixel values from the two averaged values, we need to store some *detail coefficients*, which capture the missing information. In our example, we will choose 1 for the first detail coefficient, since the average we computed is 1 less than 9 and 1 more than 7. This single number allows us to recover the first two pixels of our original four-pixel image. Similarly, the second detail coefficient is -1, since $4 + (-1) = 3$ and $4 - (-1) = 5$.

Thus, we have decomposed the original image into a lower resolution (two-pixel) version and a pair of detail coefficients. Repeating this process recursively on the averages gives the full decomposition:

Resolution	Averages	Detail coefficients
4	$\begin{bmatrix} 9 & 7 & 3 & 5 \end{bmatrix}$	
2	$\begin{bmatrix} 8 & 4 \end{bmatrix}$	$\begin{bmatrix} 1 & -1 \end{bmatrix}$
1	$\begin{bmatrix} 6 \end{bmatrix}$	$\begin{bmatrix} 2 \end{bmatrix}$

Finally, we will define the *wavelet transform* (also called the *wavelet decomposition*) of the original four-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 four-pixel image is given by

$$\begin{bmatrix} 6 & 2 & 1 & -1 \end{bmatrix}$$

The way we computed the wavelet transform, by recursively averaging and differencing coefficients, is called a *filter bank*—a process we will generalize to other types of wavelets in Part 2 of our tutorial. Note that no information has been gained or lost by this process. The original image had four 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 image's wavelet transform, rather than the image itself, has a number of advantages. One advantage of the wavelet transform is that often a large number of the detail coefficients turn out to be very small in magnitude, as in the 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 2.3, after we present the one-dimensional Haar basis functions.

†Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for computer graphics: A primer, part 1. *IEEE Computer Graphics and Applications*, 15(3):76–84, May 1995.

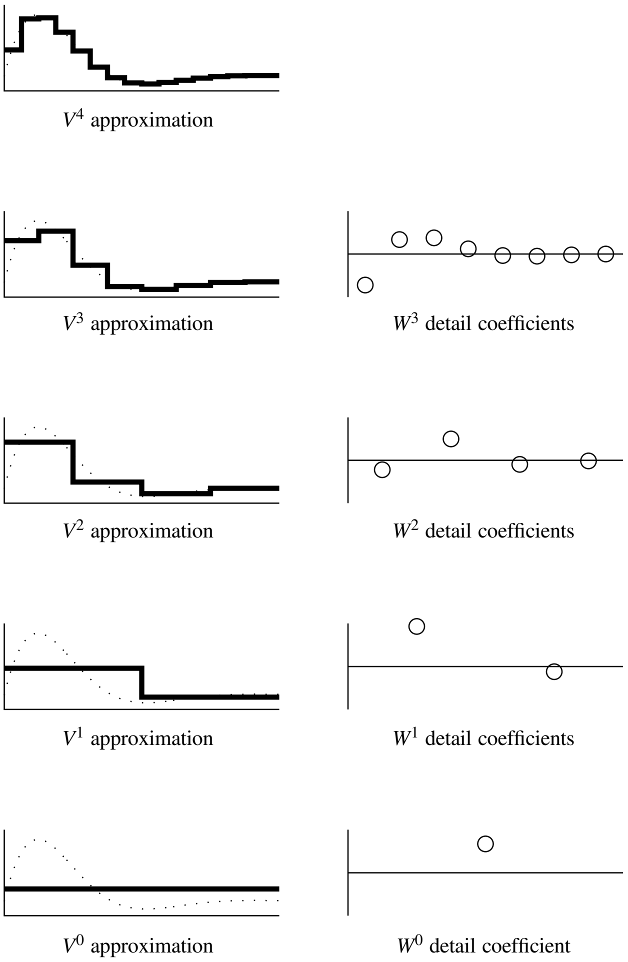


Figure 1 A sequence of decreasing-resolution approximations to a function (left), along with the detail coefficients required to recapture the finest approximation (right). Note that in regions where the true function is close to being flat, a piecewise-constant approximation works well, so the corresponding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

2.2 One-dimensional Haar wavelet basis functions

We have shown how one-dimensional images can be treated as sequences of coefficients. Alternatively, we can think of images as piecewise-constant functions on the half-open interval $[0, 1)$. To do so, we will use the concept of a *vector space* from 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 vector 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 defined on the interval $[0, 1)$ with constant pieces over each of 2^j equal subintervals.

We can now think of every one-dimensional image with 2^j pixels as 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 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,

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

The mathematical theory of *multiresolution analysis* requires this nested set of spaces V^j. We will consider this topic more thoroughly in Part 2.

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, Figure 2 shows the four box functions forming a 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. 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.

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

1. 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}.
2. 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]

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

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

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.

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:

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

[1] Some authors refer to functions with these properties as *pre-wavelets*, reserving the term "wavelet" for functions ψ_i^j that are also orthogonal to each other.

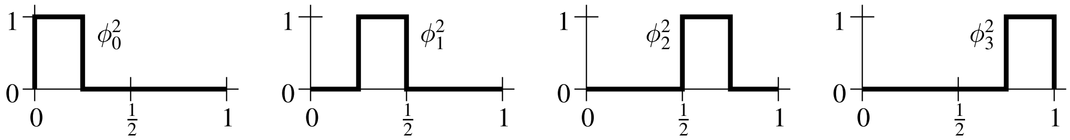


Figure 2 The box basis for V^2.

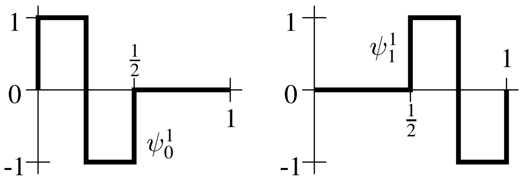
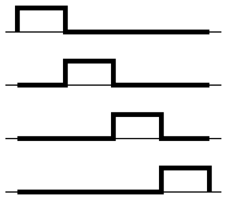


Figure 3 The Haar wavelets for W^1.

A more graphical representation is

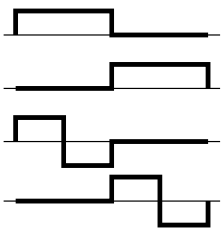
$$\begin{array}{rcrc} \mathcal{I}(x) & = & 9 & \times \\ & + & 7 & \times \\ & + & 3 & \times \\ & + & 5 & \times \end{array}$$



Note that the coefficients $c^2_0, \ldots, c^2_3$ are just the four original pixel values [9 7 3 5].

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:

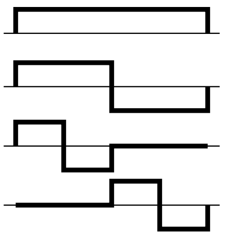
$$\begin{array}{rcl} \mathcal{I}(x) & = & c^1_0\,\phi^1_0(x) + c^1_1\,\phi^1_1(x) + d^1_0\,\psi^1_0(x) + d^1_1\,\psi^1_1(x) \\ & = & 8 \times \\ & + & 4 \times \\ & + & 1 \times \\ & + & -1 \times \end{array}$$



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:

$$\begin{array}{rcl} \mathcal{I}(x) & = & c^0_0\,\phi^0_0(x) + d^0_0\,\psi^0_0(x) + d^1_0\,\psi^1_0(x) + d^1_1\,\psi^1_1(x) \\ & = & 6 \times \\ & + & 2 \times \\ & + & 1 \times \\ & + & -1 \times \end{array}$$



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 $\phi^0_0, \psi^0_0, \psi^1_0$, and ψ^1_1 to represent 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 $\psi(x)$.

Orthogonality

The Haar basis possesses an important property known as *orthogonality*, which is not always shared by other wavelet bases. An orthogonal basis is one in which all of the basis functions, in this case $\phi^0_0, \psi^0_0, \psi^1_0, \psi^1_1, \ldots$, are orthogonal to one another. Note that orthogonality is stronger than the minimum requirement for wavelets that ψ^j_i be orthogonal to all scaling functions at the same resolution level j.

Normalization

Another property that is sometimes desirable is *normalization*. A basis function $u(x)$ is normalized if $\langle u \mid u \rangle = 1$. We can normalize the Haar basis by replacing our earlier definitions with

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

where the constant factor of $2^{j/2}$ is chosen to satisfy $\langle u \mid u \rangle = 1$ for the standard inner product. With these modified definitions, the new normalized coefficients are obtained by multiplying each old coefficient with superscript j by $2^{-j/2}$. Thus, in the example from the previous section, the unnormalized coefficients $[6\ 2\ 1\ {-1}]$ become the normalized coefficients

$$\left[\begin{array}{cccc} 6 & 2 & \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{array}\right]$$

As an alternative to first computing the unnormalized coefficients and then normalizing them, we can include normalization in the decomposition algorithm. The following two pseudocode procedures accomplish this normalized decomposition:

```
procedure DecompositionStep (C: array [1. . h] of reals)
    for i ← 1 to h/2 do
        C′[i] ← (C[2i − 1] + C[2i])/√2
        C′[h/2 + i] ← (C[2i − 1] − C[2i])/√2
    end for
    C ← C′
end procedure

procedure Decomposition (C: array [1. . h] of reals)
    C ← C/√h    (normalize input coefficients)
    while h > 1 do
        DecompositionStep (C[1. . h])
        h ← h/2
    end while
end procedure
```

Now we can work with an *orthonormal* basis, meaning one that is both orthogonal and normalized. Using an orthonormal basis turns out to be handy when compressing a function or an image, which we describe next.

2.3 Application I: Compression

The goal of compression is to express an initial set of data using some smaller set of data, either with or without loss of information. For instance, suppose we are given a function $f(x)$ expressed as a weighted sum of basis functions $u_1(x), \ldots, u_m(x)$:

$$f(x) = \sum_{i=1}^{m} c_i\, u_i(x).$$

The data set in this case consists of the coefficients $c_1, \ldots, c_m$. 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

$$\tilde{f}(x) = \sum_{i=1}^{\tilde{m}} \tilde{c}_i\, \tilde{u}_i(x)$$

such that $\tilde{m} < m$ and $\|f(x) - \tilde{f}(x)\| \leq \epsilon$ for some norm. In general, you could attempt to construct a set of basis functions $\tilde{u}_1, \ldots, \tilde{u}_{\tilde{m}}$ 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.

One form of the compression problem is to order the coefficients $c_1, \ldots, c_m$ so that for every $\tilde{m} < m$, the first $\tilde{m}$ elements of the sequence give the best approximation $\tilde{f}(x)$ 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, as is the case with the normalized Haar basis.

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

$$\tilde{f}(x) = \sum_{i=1}^{\tilde{m}} c_{\sigma(i)}\, u_{\sigma(i)}.$$

The square of the L^2 error in this approximation is

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

The last step follows from the assumption that the basis is orthonormal, so $\langle u_i \,|\, u_j \rangle = \delta_{ij}$. We conclude that to minimize this error for any given $\tilde{m}$, the best choice for σ is the permutation that sorts the coefficients in order of decreasing magnitude; that is, σ satisfies $|c_{\sigma(1)}| \geq \cdots \geq |c_{\sigma(m)}|$.

Figure 1 demonstrated how a one-dimensional function could be transformed 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. 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 4.

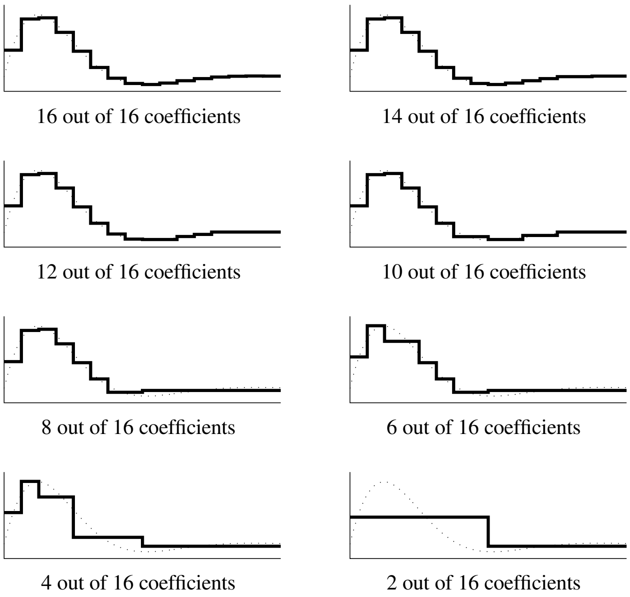


Figure 4 Coarse approximations to a function obtained using L^2 compression: detail coefficients are removed in order of increasing magnitude.

3 Wavelets in two dimensions

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

3.1 Two-dimensional Haar wavelet transforms

There are two ways we can use wavelets to transform the pixel values within an image. Each is a generalization to two dimensions of the one-dimensional wavelet transform described in Section 2.1.

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. The algorithm below computes the standard decomposition. Figure 5 illustrates each step of its operation.

```
procedure StandardDecomposition (C: array [1..h, 1..w] of reals)
    for row ← 1 to h do
        Decomposition (C[row, 1..w])
    end for
    for col ← 1 to w do
        Decomposition (C[1..h, col])
    end for
end procedure
```

The second type of two-dimensional wavelet transform, called the *nonstandard decomposition*, alternates between operations on rows

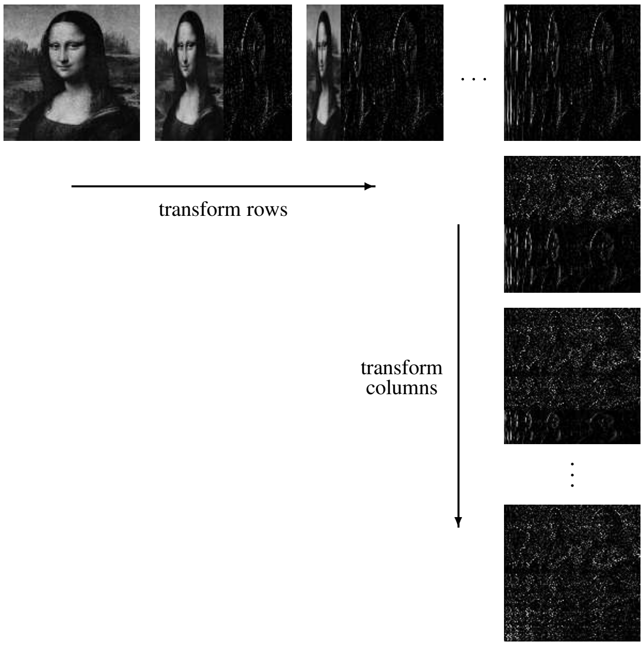


Figure 5 Standard decomposition of an image.

Figure 6 Nonstandard decomposition of an image.

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 only on the quadrant containing averages in both directions. Figure 6 shows all the steps involved in the nonstandard decomposition procedure below.

```
procedure NonstandardDecomposition(C: array [1..h, 1..h] of reals)
    C ← C/h    (normalize input coefficients)
    while h > 1 do
        for row ← 1 to h do
            DecompositionStep(C[row, 1..h])
        end for
        for col ← 1 to h do
            DecompositionStep(C[1..h, col])
        end for
        h ← h/2
    end while
end procedure
```

3.2 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 nonstandard decomposition gives coefficients for the *nonstandard 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 shown in Figure 7. Note that if we apply the standard construction to an orthonormal basis in one dimension, we get an orthonormal basis in two dimensions.

The nonstandard construction of a two-dimensional basis proceeds by first defining a two-dimensional scaling function,

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

and three wavelet functions,

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

We now denote levels of scaling with a superscript j (as we did in the one-dimensional case) and horizontal and vertical translations with a pair of subscripts k and ℓ. The nonstandard basis consists of a single coarse scaling function $\phi\phi^0_{0,0}(x, y) := \phi\phi(x, y)$ along with scales and translates of the three wavelet functions $\phi\psi$, $\psi\phi$, and $\psi\psi$:

$$\begin{aligned}
\phi\psi^j_{k\ell}(x, y) &:= 2^j\,\phi\psi(2^j x - k, 2^j y - \ell) \\
\psi\phi^j_{k\ell}(x, y) &:= 2^j\,\psi\phi(2^j x - k, 2^j y - \ell) \\
\psi\psi^j_{k\ell}(x, y) &:= 2^j\,\psi\psi(2^j x - k, 2^j y - \ell).
\end{aligned}$$

The constant 2^j normalizes the wavelets to give an orthonormal basis. The nonstandard construction results in the basis for V^2 shown in Figure 8.

We have presented both the standard and nonstandard approaches to wavelet transforms and basis functions because both have advantages. The standard decomposition of an image is appealing because it simply requires performing one-dimensional transforms on all rows and then on all columns. On the other hand, it is slightly more efficient to compute the nonstandard decomposition. For an $m \times m$ image, the standard decomposition requires $4(m^2 - m)$ assignment operations, while the nonstandard decomposition requires only $\frac{8}{3}(m^2 - 1)$ assignment operations.

Another consideration is the *support* of each basis function, meaning the portion of each function's domain where that function is nonzero. All nonstandard Haar basis functions have square supports,

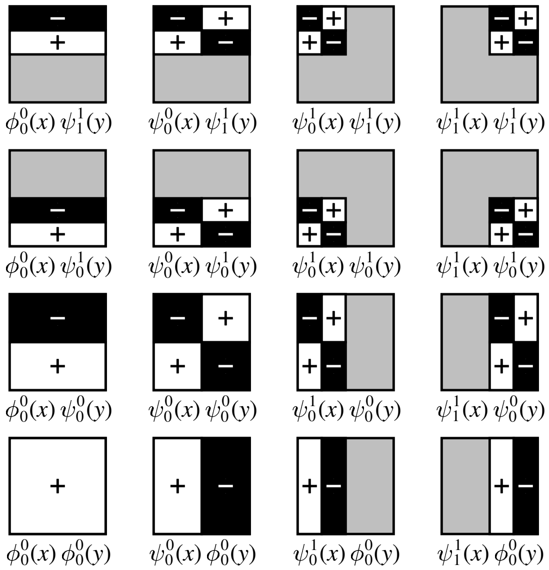


Figure 7 Standard construction of a two-dimensional Haar wavelet basis for V^2. In the unnormalized case, functions are +1 where plus signs appear, -1 where minus signs appear, and 0 in gray regions.

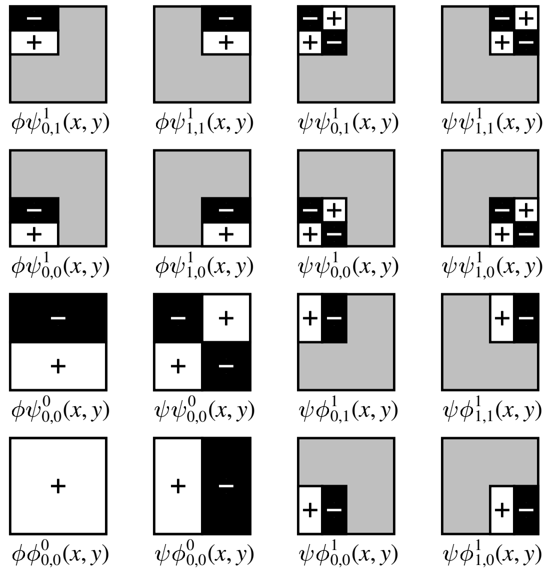


Figure 8 Nonstandard construction of a two-dimensional Haar wavelet basis for V^2.

while some standard basis functions have nonsquare supports. Depending upon the application, one of these choices may be preferable to the other.

3.3 Application II: Image compression

We defined compression in Section 2.3 as the representation of a function using fewer basis function coefficients than were originally given. The method we discussed for one-dimensional functions applies equally well to images, which we treat as the coefficients corresponding to a two-dimensional piecewise-constant basis. The approach presented here is only introductory; for a more complete treatment of wavelet image compression, see the article by DeVore *et al.* [6].

We can summarize wavelet image compression using the L^2 norm in three steps:

1. Compute coefficients $c_1, \ldots, c_m$ representing an image in a normalized two-dimensional Haar basis.
2. Sort the coefficients in order of decreasing magnitude to produce the sequence $c_{\sigma(1)}, \ldots, c_{\sigma(m)}$.
3. Starting with $\tilde{m} = m$, find the smallest $\tilde{m}$ for which $\sum_{i=\tilde{m}+1}^{m} (c_{\sigma(i)})^2 \leq \epsilon^2$, where ϵ is the allowable L^2 error.

The first step is accomplished by applying either of the two-dimensional Haar wavelet transforms described in Section 3.1, being 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 uses a binary search strategy to find a threshold below which coefficient sizes are deemed negligible. The procedure takes as input a one-dimensional array of coefficients C (with each coefficient corresponding to a two-dimensional basis function) and an error tolerance ϵ. For each guess at a threshold τ, the algorithm computes the square of the L^2 error that would result from discarding coefficients smaller in magnitude than τ. This squared error s is compared to ϵ^2 at each iteration to decide whether the binary search should continue in the upper or lower half of the current interval. The algorithm halts when the current interval is so narrow that the number of coefficients to be discarded no longer changes.

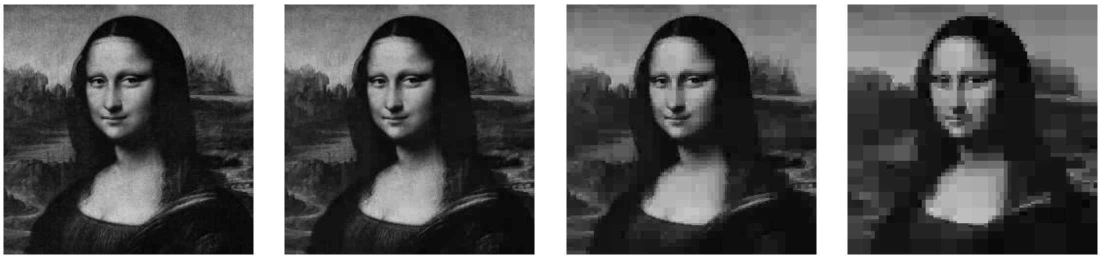
procedure *Compress*(C: **array** $[1..m]$ **of reals**; ϵ: **real**)
  $\tau_{\min} \leftarrow \min \{ |C[i]| \}$
  $\tau_{\max} \leftarrow \max \{ |C[i]| \}$
  do
    $\tau \leftarrow (\tau_{\min} + \tau_{\max})/2$
    $s \leftarrow 0$
    for $i \leftarrow 1$ **to** m **do**
      if $|C[i]| < \tau$ **then** $s \leftarrow s + (C[i])^2$
    end for
    if $s < \epsilon^2$ **then** $\tau_{\min} \leftarrow \tau$ **else** $\tau_{\max} \leftarrow \tau$
  until $\tau_{\min} \approx \tau_{\max}$
  for $i \leftarrow 1$ **to** m **do**
    if $|C[i]| < \tau$ **then** $C[i] \leftarrow 0$
  end for
end procedure

This binary search algorithm 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.

DeVore *et al.* [6] suggest that the L^1 norm is best suited to the task of image compression. Here is a pseudocode fragment for a "greedy" L^1 compression scheme:

for each pixel (x, y) **do**
  $\delta[x, y] \leftarrow 0$
end for
for $i \leftarrow 1$ **to** m **do**
  $\delta' \leftarrow \delta$ + error from discarding $C[i]$
  if $\sum_{x,y} |\delta'[x, y]| < \epsilon$ **then**
    discard coefficient $C[i]$
    $\delta \leftarrow \delta'$
  end if
end for

Note that this algorithm's results depend on the order in which coefficients are visited. Different images (and degrees of compression) may be obtained from varying this order—for example, by starting with the finest scale coefficients, rather than the smallest coefficients. You could also run a more sophisticated constrained optimization procedure to select the minimum number of coefficients subject to the error bound.



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

Figure 9 L^2 wavelet image compression: The original image (a) can be represented using (b) 19% of its wavelet coefficients, with 5% relative L^2 error; (c) 3% of its coefficients, with 10% relative L^2 error; and (d) 1% of its coefficients, with 15% relative L^2 error.

4 Conclusion

We have described Haar wavelets in one and two dimensions as well as how to use them for compressing functions and images. Part 2 of this primer will continue this exposition by presenting the mathematical framework of multiresolution analysis. We will also develop a class of wavelets based on endpoint-interpolating B-splines, and describe how to use them for multiresolution curve and surface editing.

Acknowledgments

We wish to thank Ronen Barzel, Steven Gortler, Michael Shantzis, and the anonymous reviewers for many helpful comments. 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 Royalty Research Fund (65-9731), and by industrial gifts from Adobe, Aldus, Microsoft, and Xerox.

References

[1] Deborah Berman, Jason Bartell, and David Salesin. Multiresolution painting and compositing. In *Proceedings of SIGGRAPH 94*, pages 85–90. ACM, New York, 1994.

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

[3] Per H. Christensen, Dani Lischinski, Eric J. Stollnitz, and David H. Salesin. Clustering for glossy global illumination. *ACM Transactions on Graphics*, 1996 (to appear).

[4] Per H. Christensen, Eric J. Stollnitz, David H. Salesin, and Tony D. DeRose. Wavelet radiance. In G. Sakas, P. Shirley, and S. Müller, editors, *Photorealistic Rendering Techniques*, pages 295–309. Springer-Verlag, Berlin, 1995.

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

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

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

[8] Steven J. Gortler and Michael F. Cohen. Hierarchical and variational geometric modeling with wavelets. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, pages 35–42. ACM, New York, 1995.

[9] Steven J. Gortler, Peter Schröder, Michael F. Cohen, and Pat Hanrahan. Wavelet radiosity. In *Proceedings of SIGGRAPH 93*, pages 221–230. ACM, New York, 1993.

[10] Charles E. Jacobs, Adam Finkelstein, and David H. Salesin. Fast multiresolution image querying. In *Proceedings of SIGGRAPH 95*, pages 277–286. ACM, New York, 1995.

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

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

[13] 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.

[14] David Meyers. Multiresolution tiling. *Computer Graphics Forum*, 13(5):325–340, December 1994.

[15] Peter Schröder, Steven J. Gortler, Michael F. Cohen, and Pat Hanrahan. Wavelet projections for radiosity. *Computer Graphics Forum*, 13(2):141–151, June 1994.

[16] 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 Linear algebra review

The mathematics of wavelets rely heavily on fundamental ideas from linear algebra. This appendix reviews a few important ideas.

A.1 Vector spaces

The starting point for linear algebra is the notion of a *vector space*. A vector space (over the reals) can be loosely defined as a collection V of elements where

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 $\boldsymbol{0} \in V$ such that
   - for all $u \in V$, $0u = \boldsymbol{0}$, and
   - for all $u \in V$, $\boldsymbol{0} + u = u$.
3. Other axioms (omitted here) hold true, most of which are necessary to guarantee that multiplication and addition behave as expected.

The elements of a vector space V are called *vectors*, and the element $\boldsymbol{0}$ 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

$$c_1 u_1 + c_2 u_2 + \cdots = \boldsymbol{0} \quad \text{if and only if} \quad c_1 = c_2 = \cdots = 0.$$

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 $c_1, c_2, \ldots$. The vectors in a basis for V are said to *span* V. Intuitively speaking, linear independence means that the vectors are not redundant, and a basis consists of a minimal complete set of 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 three-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 $\langle \cdot \mid \cdot \rangle$ on a vector space V is any map from $V \times V$ to $\mathbb{R}$ that is

1. symmetric: $\langle u \mid v \rangle = \langle v \mid u \rangle$,
2. bilinear: $\langle au + bv \mid w \rangle = a\langle u \mid w \rangle + b\langle v \mid w \rangle$, and
3. positive definite: $\langle u \mid u \rangle > 0$ for all $u \neq \boldsymbol{0}$.

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

$$\langle (a_1, a_2, a_3) \mid (b_1, b_2, b_3) \rangle := a_1 b_1 + a_2 b_2 + a_3 b_3 \qquad (1)$$

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:

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

The standard inner product can also be generalized to include a positive weight function $w(x)$:

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

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 $\langle u \mid v \rangle = 0$. 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 $\|u\| := \langle u \mid u \rangle^{1/2}$. If we are working with a function space such as $C[0, 1]$, we ordinarily use one of the L^p norms, defined as

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

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

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

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

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

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 (1).