HAAR: THE SIMPLEST WAVELET BASIS


The Haar basis is the simplest wavelet basis. In this chapter, we will begin by examining how a one-dimensional function can be decomposed using Haar wavelets. We will then look at the Haar basis functions in detail and see how Haar wavelet decomposition can be used for compression. Later, in the following three chapters, we'll explore some applications of the Haar basis: image compression, image editing, and image querying.

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:

\[ 9 \quad 7 \quad 3 \quad 5 \]
This image can be represented in the Haar basis, the simplest wavelet basis, by computing a wavelet transform as follows. Start by averaging the pixels together, pairwise, to get a new lower-resolution image with these pixel values:

\[ [8, 4] \]

Clearly, some information has been lost in this averaging and down-sampling 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 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 \(4 - (-1) = 5\).

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

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Averages</th>
<th>Detail Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>[9 7 3 5]</td>
<td>[1 -1]</td>
</tr>
<tr>
<td>2</td>
<td>[8 4]</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Averages</th>
<th>Detail Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>[9 7 3 5]</td>
<td>[1 -1]</td>
</tr>
<tr>
<td>2</td>
<td>[8 4]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>[6]</td>
<td></td>
</tr>
</tbody>
</table>

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

\[ [6 2 1 -1] \]

2.2 One-dimensional Haar basis functions

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 Chapter 7. 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 wavelet transform of the image, 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 larger example of Figure 2.1. 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 2.4, once we have presented the one-dimensional Haar basis functions.

2.2 One-dimensional Haar 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 \([0, 1)\). (A half-open interval \([a, b)\) contains all values of \(x\) in the range \(a \leq x < b\).) In this new treatment, we will use the concept of a “vector space” from linear algebra. (A refresher on linear algebra can be found in Appendix A.)

A vector space \(V\) is basically just a collection of “things” (called vectors in this context) for which addition and scalar multiplication are defined. Thus, you can add two vectors, scale a vector by some constant, and so forth. (The full list of axioms can be found in Appendix A.1.)

Until now, we have been thinking of images as sequences of coefficients; let’s instead think of them as functions. For example, we can consider a one-pixel image to be a function that is constant over the entire interval \([0, 1)\). Since addition and scalar multiplication of functions are well defined, we can then think of each constant function over the interval \([0, 1)\) as a vector, and we’ll let \(V^0\) denote the vector space of all such functions. Similarly, a two-pixel image is a function having 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-sized subintervals.
2.2 One-dimensional Haar basis functions

\[ V^0 \subset V^1 \subset V^2 \subset \cdots \]

This nested set of spaces \( V^i \) is a necessary ingredient for the mathematical theory of multiresolution analysis, a topic we will consider more thoroughly in Chapter 7.

Now we need to define a basis for each vector space \( V^i \). A basis for a vector space is defined formally in Appendix A.2. Roughly speaking, a basis consists of a minimum set of vectors from which all other vectors in the vector space can be generated through linear combinations. The basis functions for the spaces \( V^i \) are called scaling functions and are usually denoted by the symbol \( \phi \). A simple basis for \( V^0 \) is given by the set of scaled and translated box functions:

\[ \phi_i^j(x) := \phi(2^j x - i) \quad i = 0, \ldots, 2^j - 1 \]

where

\[ \phi(x) := \begin{cases} 
1 & \text{for } 0 \leq x < 1 \\
0 & \text{otherwise}
\end{cases} \]

As an example, the four box functions forming a basis for \( V^2 \) are shown in Figure 2.2.

The support of a function refers to the region of the parameter domain over which the function is nonzero. For example, the support of \( \phi_0^0(x) \) is \( [0, 1/4) \). Functions that are supported over a bounded interval are said to have compact support. Note that all of the box functions are compactly supported.

The next step in building a multiresolution analysis is to choose an inner product defined on the vector spaces \( V^i \) (see Appendix A.3 for a formal definition of inner products). For our running example, the "standard" inner product will do quite well.
Two vectors \( u \) and \( v \) are said to be orthogonal under a chosen inner product if \( \langle u \mid v \rangle = 0 \). We can now define a new vector space \( W_i \) as the orthogonal complement of \( V_i \) in \( V^{m_i} \). In other words, \( W_i \) is the space of all functions in \( V^{m_i} \) that are orthogonal to all functions in \( V_i \) under the chosen inner product.

A collection of linearly independent functions \( \psi_i \) spanning \( W_i \) are called wavelets. These basis functions have two important properties:

1. The basis functions \( \psi_i \) of \( W_i \), together with the basis functions \( \phi_i \) of \( V_i \), form a basis for \( V^{m_i} \).
2. Every basis function \( \psi_i \) of \( W_i \) is orthogonal to every basis function \( \phi_j \) of \( V_i \) under the chosen inner product.

**Remark:** Later, in Chapter 7, we'll look at ways in which the definitions of the complement spaces \( W_i \) and the wavelets \( \psi_i \) above can either be made more strict or more relaxed. For example, some authors refer to the functions defined as wavelets above as pre-wavelets, reserving the term wavelets for functions \( \psi_i \) that are orthogonal to each other as well.

Informally, we can think of the wavelets in \( W_i \) as a means of representing the parts of a function in \( V^{m_i} \) that cannot be represented in \( V_i \). 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

\[
\psi_i(x) := \psi(2^i / x - i) \quad i = 0, \ldots, 2^j - 1
\]

where

\[
\psi(x) = \begin{cases} 
1 & \text{for } 0 \leq x < 1/2 \\
-1 & \text{for } 1/2 \leq x < 1 \\
0 & \text{otherwise}
\end{cases}
\]

Figure 2.3 shows the two Haar wavelets spanning \( W^j \).
These four coefficients should look familiar as well. Finally, we'll rewrite \( \mathcal{R}(x) \) as a sum of basis functions in \( V^0, W^0, \) and \( W^1 \):

\[
\mathcal{R}(x) = c_0 \phi_0^0(x) + d_0 \psi_0^0(x) + d_1 \phi_1(x) + d_1 \psi_1(x)
\]

\[
= 6 \times \begin{array}{c}
\end{array} + 2 \times \begin{array}{c}
\end{array} + 1 \times \begin{array}{c}
\end{array} + (-1) \times \begin{array}{c}
\end{array}
\]

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 \( \phi_0^0, \psi_0^0, \phi_1, \psi_1 \) 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 four functions as well as even narrower versions of the wavelet \( \psi(x) \).

### 2.3 Orthogonality and normalization

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 \( \phi_0, \psi_0, \phi_1, \psi_1, \ldots \), are orthogonal to one another. Note that orthogonality is stronger than the requirement in the definition of wavelets that \( \psi \) be orthogonal to all scaling functions at the same hierarchy level \( j \).

Another property of some wavelet bases is normalization. A basis function \( u(x) \) is normalized if \( \langle u, u \rangle = 1 \). We can normalize the Haar basis by replacing our earlier definitions with:

\[
\phi_j(x) := \sqrt{2^j} \phi(x - i) \\
\psi_j(x) := \sqrt{2^j} \psi(x - i)
\]

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

\[
\begin{bmatrix}
6 & 1 & -1 \\
\sqrt{2} & \sqrt{2} & \sqrt{2}
\end{bmatrix}
\]

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

**procedure Decomposition(c: array [1 \ldots 2^l] of reals)**

\[
c \leftarrow c/\sqrt{2^l} \quad \text{(normalize input coefficients)} \\
g \leftarrow 2^l
\]

while \( g \geq 2 \)

\[
\text{DecompositionStep}([c_{[1 \ldots g]}])
\]

\[
g \leftarrow g/2
\]

end while

**end procedure**

**procedure DecompositionStep(c: array [1 \ldots 2^l] of reals)**

\[
\text{for } i \leftarrow 1 \text{ to } 2^l/2 \text{ do}
\]

\[
c'[i] \leftarrow (c[2i-1] + c[2i+1])/\sqrt{2}
\]

\[
c'[2^l/2 + i] \leftarrow (c[2i-1] - c[2i+1])/\sqrt{2}
\]

end for

\[
c \leftarrow c'
\]

**end procedure**

Of course, after we obtain a wavelet decomposition, we need to be able to reconstruct the original data. The following two pseudocode procedures do just that:

**procedure Reconstruction(c: array [1 \ldots 2^l] of reals)**

\[
g \leftarrow 2
\]

while \( g \leq 2^l \)

\[
\text{ReconstructionStep}([c_{[1 \ldots g]}])
\]

\[
g \leftarrow 2g
\]

end while

\[
c \leftarrow c/\sqrt{2^l} \quad \text{(undo normalization)}
\]

**end procedure**
procedure ReconstructionStep(c: array [1 . . . 2^l] of reals)
    for i ← 1 to 2^l/2 do
        c'[2i - 1] ← (c[i] + c[2^l/2 + i])/√2
        c'[2i] ← (c[i] - c[2^l/2 + i])/√2
    end for
    c ← c'
end procedure

The above pseudocode procedures allow us to work with an orthonormal basis: one that
is both orthogonal and normalized. As we will see in the next section, using an orthonormal
basis turns out to be handy when compressing a function or an image.

2.4 Wavelet 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), . . . , 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, . . . , c_m. We would like to find
a function approximating f(x) but requiring fewer coefficients, perhaps by using a different ba-
sis. That is, given a user-specified error tolerance ε (for lossless compression, ε = 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 \( \| f(x) - \tilde{f}(x) \| \leq \varepsilon \) for some norm (see Appendix A.4 for more on norms). In
general, one could attempt to construct a set of basis functions \( \hat{u}_1, \ldots, \hat{u}_2 \) that would provide
a good approximation with few coefficients. We will focus instead on the simpler prob-
lem of finding a good approximation in a fixed basis. Note that here and elsewhere in
the book, when we discuss compression, we are concentrating on reducing the number of coeffi-
cients needed to represent a function—and not on the equally challenging problem of encoding
and storing the necessary information in the fewest possible bits.

One form of the compression problem is to order the coefficients c_1, . . . , c_m so that for
every \( \hat{m} < m \), the first \( \hat{m} \) 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 \( \pi(i) \) be a permutation of 1, . . . , m and let \( \tilde{f}(x) \) be a function that uses the coefficients
Corresponding to the first \( \hat{m} \) numbers of the permutation \( \pi(i) \):

\[ \tilde{f}(x) = \sum_{i=1}^{\hat{m}} c_{\pi(i)} \hat{u}_{\pi(i)}(x) \]

The square of the \( L^2 \) error in this approximation is given by

\[ \| f(x) - \tilde{f}(x) \|^2 = \langle f(x) - \tilde{f}(x) \mid f(x) - \tilde{f}(x) \rangle \]

\[ = \sum_{i=1}^{m} \sum_{j=1}^{m} \tilde{c}_i \tilde{c}_j \langle \hat{u}_{\pi(i)} \mid \hat{u}_{\pi(j)} \rangle \]

\[ = \sum_{i=1}^{\hat{m}} \sum_{j=1}^{\hat{m}} \tilde{c}_{\pi(i)} \tilde{c}_{\pi(j)} \delta_{\pi(i) \pi(j)} \]

\[ = \sum_{i=1}^{\hat{m}} \tilde{c}_{\pi(i)}^2 \]

The last step follows from the assumption that the basis is orthonormal, so \( \langle \hat{u}_i \mid \hat{u}_j \rangle = \delta_{ij} \). The
above result indicates that the square of the overall \( L^2 \) error is just the sum of the squares of all
the coefficients we choose to leave out. We conclude that in order to minimize this error for
any given \( \hat{m} \), the best choice for \( \pi(i) \) is the permutation that sorts the coefficients in order of
decreasing magnitude; that is, \( \pi(i) \) satisfies

\[ |c_{\pi(1)}| \geq \cdots \geq |c_{\pi(m)}| \]

Figure 2.1 demonstrated how a one-dimensional function could be transformed into coeffi-
cients 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 2.4.

**FIGURE 2.4** Coarse approximations to a function obtained using $L^2$ compression: detail coefficients are removed in order of increasing magnitude.

In preparation for image compression, we need to generalize Haar wavelets to two dimensions. First, we 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. These tools will enable us to describe image compression as an application of wavelets.

### 3.1 Two-dimensional Haar wavelet transforms

There are two common ways in which wavelets can be used to transform the pixel values within an image. Each of these transformations is a two-dimensional generalization of the one-dimensional wavelet transform described in Section 2.1.

The first transform is called the standard decomposition $[5]$. To obtain the standard decomposition 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.
3.1 Two-dimensional Haar wavelet transforms

\[ \text{Decomposition}(c[1 \ldots 2^i, \text{col}]) \]
\[ \text{end for} \]
\[ \text{end procedure} \]

The corresponding reconstruction algorithm simply reverses the steps performed during decomposition:

\[ \text{procedure StandardReconstruction}(c; \text{array } [1 \ldots 2^i, 1 \ldots 2^i] \text{ of reals}) \]
\[ \text{for col} \leftarrow 1 \text{ to } 2^i \text{ do} \]
\[ \text{Reconstruction}(c[1 \ldots 2^i, \text{col}]) \]
\[ \text{end for} \]
\[ \text{for row} \leftarrow 1 \text{ to } 2^i \text{ do} \]
\[ \text{Reconstruction}(c[\text{row}, 1 \ldots 2^i]) \]
\[ \text{end for} \]
\[ \text{end procedure} \]

The second type of two-dimensional wavelet transforms, called the nonstandard decomposition \cite{5}, 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 only on the quadrant containing averages in both directions. Figure 3.2 shows all the steps involved in the nonstandard decomposition procedure below.

\[ \text{procedure NonstandardDecomposition}(c; \text{array } [1 \ldots 2^i, 1 \ldots 2^i] \text{ of reals}) \]
\[ c \leftarrow c/2^i \quad \text{(normalize input coefficients)} \]
\[ g \leftarrow 2^i \]
\[ \text{while } g \geq 2 \text{ do} \]
\[ \text{for row} \leftarrow 1 \text{ to } g \text{ do} \]
\[ \text{DecompositionStep}(c[\text{row}, 1 \ldots g]) \]
\[ \text{end for} \]
\[ \text{for col} \leftarrow 1 \text{ to } g \text{ do} \]
\[ \text{DecompositionStep}(c[1 \ldots g, \text{col}]) \]
\[ \text{end for} \]
\[ g \leftarrow g/2 \]
\[ \text{end while} \]
\[ \text{end procedure} \]
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 of a two-dimensional basis. Similarly, the nonstandard decomposition gives coefficients for the nonstandard construction of basis functions [5].

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 $V^1$, we get the two-dimensional basis for $V^2$ shown in Figure 3.3. 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,

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

and three wavelet functions,

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

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

$$\psi(x, y) := \psi(x) \psi(y)$$

Here is the pseudocode to perform the nonstandard reconstruction:

```plaintext
procedure NonstandardReconstruction(c: array [1..2^l, 1..2^l] of reals)
g ← 2
while g ≤ 2^l do
    for col ← 1 to g do
        ReconstructionStep(c[1..g, col])
    end for
    for row ← 1 to g do
        ReconstructionStep(c[row, 1..g])
    end for
    g ← 2g
end while
c ← 2/c (undo normalization)
end procedure
```
3. Two-dimensional Haar basis functions

We have presented both the standard and nonstandard 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 nonstandard decomposition of an image. For an $m \times m$ image, the standard decomposition requires $4(m^2 - m)$ assignment operations, while the nonstandard decomposition requires only $3(m^2 - 1)$ 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 of the nonstandard Haar basis functions have square supports, while some of the standard basis functions have nonsquare supports. Depending upon the application, one of these choices may be preferable to the other.
3.3 Wavelet image compression

We defined compression in Section 2.4 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. [27]. Once again, we note that we are dealing only with the transformation and quantization of coefficients and not with how they are encoded.

Wavelet image compression using the $L^2$ norm can be summarized 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 $c_{(1)}, \ldots, c_{(m)}$.
3. Given an allowable $L^2$ error $\varepsilon^2$ and starting with $\hat{n} = m$, find the smallest $\hat{n}$ for which

$$\sum_{k=1}^{n} (\delta_k)^2 \leq \varepsilon^2$$

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, and any standard search can be used for the third step. However, for large images sorting becomes exceedingly slow. The pseudocode below outlines a more efficient method of accomplishing steps 2 and 3, which uses a binary search strategy to find a threshold $\tau$ below which coefficients can be truncated.

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 $\varepsilon$. For each guess at a threshold $\tau$, the algorithm computes the square of the $L^2$ error that would result from discarding coefficients smaller in magnitude than $\tau$. This squared error $s$ is compared to $\varepsilon^2$ 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.

```plaintext
procedure Compress(c: array [1..m] of reals; ε: real)
    τ_min ← min(|c[i]|)
    τ_max ← max(|c[i]|)
    do
        τ ← (τ_min + τ_max)/2
        s ← 0
        for i ← 1 to m do
            if |c[i]| < τ then s ← s + |c[i]|^2
        end for
        if s < ε^2 then τ_min ← τ else τ_max ← τ
    until τ_min = τ_max
    for i ← 1 to m do
        if |c[i]| < τ then c[i] ← 0
    end for
end procedure
```

The binary search algorithm given above was used to produce the images in Figure 3.5. These images demonstrate the high compression ratios wavelets offer as well as some of the artifacts they introduce.

DeVore et al. [27] 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, which works by accumulating in a two-dimensional array $\Delta(x, y)$ the error introduced by discarding a coefficient and checking whether this error has exceeded a user-specified threshold:

```plaintext
for each pixel (x, y) do
    Δ(x, y) ← 0
end for
for i ← 1 to m do
    Δ' ← Δ + error from discarding c[i]
    if $\sum_{x,y} |\Delta(x, y)| < \varepsilon$ then
        c[i] ← 0
        Δ ← Δ'
    end if
end for
```

The algorithm proceeds by sorting the coefficients in descending order of magnitude and successively removing the coefficient with the largest magnitude. The error introduced by the removal is added to the $\Delta(x, y)$ channel at the pixel corresponding to that coefficient. The process continues until the total error introduced by the removal of the current coefficient is less than the user-specified threshold $\varepsilon$. When this condition is satisfied, the current coefficient is discarded, and the error is added to the $\Delta(x, y)$ channel at the pixel corresponding to that coefficient. The process continues until all coefficients have been processed.
3.5 Summary

Larger error in the $Q$ component of the compressed image, thereby increasing the amount of compression. (The same principle allows U.S. color television signals to be broadcast with bandwidths of 4 MHz for $Y$, 1.5 MHz for $I$, and 0.6 MHz for $Q$.)

3.5 Summary

In this and the previous chapter, we have described Haar wavelets in one and two dimensions, as well as how they can be used to compress functions and images. The Haar basis is also useful for image editing and querying, as described in the next two chapters, as well as for global illumination, as described in Chapter 13.

The theoretical exposition of wavelets will continue in Chapters 6 and 7, which present the theory of subdivision curves and show how this theory can be used in developing a more complete mathematical framework for multiresolution analysis.