

# Image Compression Using Wavelets

Peter Nacken

**Abstract.** A wavelet analysis of a signal can separate information present at different scales. Moreover, wavelet bases are orthonormal. Therefore, the wavelet transformation can be used to remove redundancy from a signal. In this paper, it is demonstrated how the wavelet transform can be used for image compression.

## §1 Introduction to data compression

Image compression techniques and, in general, data compression techniques are techniques for storing or sending images using as few bits as possible for encoding a complete image. A compressed image can either be exactly equal to the original image, or differ from it in a limited and controlled way. An image typically consists of  $256 \times 256$  pixels, each having an 8-bit grey value. Images produced by satellites can be much larger and contain more bits per pixel. The number of possible images of the mentioned format is  $(2^8)^{256 \times 256}$ . A straightforward coding of such an image would require 524288 bits. The images one usually comes across can be coded in a much smaller number of bits. This is due to the fact that natural images contain a large amount of structure, making pixel values predictable.

The idea that predictable information can be coded more efficiently than random information is due to Shannon [5].

Predictable information can be coded in fewer bits than unpredictable information. This can be seen from a simple example. Suppose that one wants to encode a message which is either A, B, C, or D. There are four possible messages, and these can be coded in two bits per message, for example using the identifications  $A \leftrightarrow 00$ ,  $B \leftrightarrow 01$ ,  $C \leftrightarrow 10$ ,  $D \leftrightarrow 11$ .

Suppose that it is known that some messages are to be sent more frequently than others, for example:

$A$  with relative frequency  $1/2$ ,  
 $B$  with relative frequency  $1/4$ ,  
 $C$  with relative frequency  $1/8$ ,  
 $D$  with relative frequency  $1/8$ .

Message  $A$  will be sent relatively often. Therefore, it pays to give  $A$  a shorter code, while giving the messages which occur less frequently a longer code. A possible coding scheme is  $A \leftrightarrow 0$ ,  $B \leftrightarrow 10$ ,  $C \leftrightarrow 110$ ,  $D \leftrightarrow 111$ . Note that no code word is equal to the first part of another code word. Therefore it is possible to send code words of different length without separators. The average number of bits per message is now:

$$\frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 = 1.75.$$

Although the length of some code words is longer than 2, the net effect of the alternative coding scheme is the use of fewer bits per message. It is of course necessary that the receiver knows the code, but this has to be sent only once. Therefore this coding is efficient if a large number of messages is to be sent.

Suppose that there are  $n$  possible messages having relative frequencies  $(p_1, \dots, p_n)$ . It can be shown [5] that the number of bits per message required by any coding scheme is at least  $H(p_1, \dots, p_n)$ , where

$$H(p_1, \dots, p_n) = - \sum_{i=1}^n p_i \log p_i,$$

( $\log$  is the base-2 logarithm). If each relative frequency is of the form  $p/2^q$ , a coding scheme can be designed in which each message has its own code word such that the optimal compression rate is indeed achieved. If the relative frequencies are not of such a special form, the optimal compression rate can not be achieved if each message is to be given its own code word, but it can be approximated arbitrarily close if messages are sent in groups and each group is given its own code word. The quantity  $H$  is called the *entropy* of the sequence  $p_1, \dots, p_n$ . The entropy is maximal if all messages are equally probable, *i.e.*,  $p_i = 1/n$  for each  $i$ . If the distribution of the messages is very peaked, the entropy is small.

If some grey values occur more often than others in an image, a coding scheme as described above can be used to compress the image. The design of the code is based on the relative frequencies of the grey values, which can be derived from the histogram. The compression achieved by this method is not very large. The bit rate for some typical images is about 7.5 bits per pixel (bpp) instead of 8 bpp if no compression is done.

## §2 Image decomposition using multiresolution analysis

For most classes of images, individual pixel values can be predicted not only by considering the histogram of the complete image, but also by looking at the values of neighbouring pixels. A much more efficient coding scheme can therefore be designed by taking into account the spatial correlations which occur in natural images. In order to use the spatial correlations, the image must be transformed or decomposed in such a way that the correlation between the parameters describing the image is removed. **A good image compression scheme requires a good image decomposition scheme.** Image decomposition can be done as follows: the image is split into a low resolution part, which can be described by a smaller number of samples than the original image, and a difference signal, which describes the difference between the low resolution image and the actual image to be coded. Due to the correlations in natural images, the low resolution version will be a good prediction of the true image; the difference signal will have a histogram which is peaked around zero. As a result of this, the difference signal can be coded with a relatively small number of bits per pixel. The low resolution image can be described by a smaller number of samples than the original image. Thus the total number of bits required to encode the image is smaller than the original number of bits to describe the whole image. The low resolution image will still contain spatial correlations. Therefore, this image is also decomposed into a low resolution image and a difference image, thus making more efficient coding possible. This decomposition is repeated several times, such that a hierarchical image decomposition is created.

In the classical image decomposition scheme of Burt and Adelson [1], a low resolution image is calculated by applying a low pass filtering, followed by subsampling. Only a quarter of the pixel values in the filtered image are stored. Thus, the low resolution image has only half the size of the original image. This reduced image is expanded to the original image size and subtracted from the original image. The result is the detail image. The detail image has the same size as the original image. Therefore, the detail image and the reduced image together contain more pixels than the original image. Yet the entropy of the detail images is so low (for natural images) that compression can be achieved with this scheme.

We now come to the description of an efficient decomposition scheme based on multiresolution wavelet bases. For a comprehensive discussion of multiresolution wavelet techniques, the reader is referred to Heijmans [3]. First the one dimensional case is described. The two dimensional case is a straightforward extension of the one dimensional case. Image compression using wavelets is described by Mallat [4] and has been worked out by Daubechies [2]. We will follow the treatment and notation of Daubechies.

Let  $\phi$  be the generator of a multiresolution wavelet basis. We put:

$$\phi_{mn}(x) = 2^{-m/2} \phi(2^{-m}x - n).$$

The spaces  $V_m = \text{span}\{\phi_{mn} | n \in \mathbb{Z}\}$  correspond with the different resolution

levels of our decomposition. There is a function  $\psi$  such that the space  $W_m = \text{span}\{\psi_{mn}|n \in \mathbb{Z}\}$  with  $\psi_{mn} = 2^{-m/2}\psi(2^{-m}x - n)$  satisfies

$$V_{m-1} = V_m \oplus W_m.$$

Let  $P_m$  and  $Q_m$  denote the orthogonal projections on  $V_m$  and  $W_m$ , respectively. Let the sequence  $(c_n)_{n \in \mathbb{Z}} \in \ell^2(\mathbb{Z})$  be the signal we want to compress. It will be convenient to define a sequence  $(c_n^0)_{n \in \mathbb{Z}}$  with  $c_n^0 = c_n$ . With this sequence we associate a function  $f \in V_0$  defined by:

$$f(x) = \sum_n c_n^0 \phi_{0n}(x).$$

A multiresolution analysis is applied to  $f$ . We can write:

$$f = P_1 f + Q_1 f.$$

The first term is the low resolution representation of  $f$ , contained in  $V_1$ , whereas the second term is the difference signal, contained in  $W_1$ . We can write:

$$P_1 f = \sum_k c_k^1 \phi_{1k}.$$

Then  $c_k^1 = \langle P_1 f, \phi_{1k} \rangle = \langle f, \phi_{1k} \rangle = \sum_n c_n^0 \langle \phi_{0n}, \phi_{1k} \rangle = \sum_n c_n^0 h_{n-2k}$  where

$$h_n = 2^{-1/2} \int \phi\left(\frac{x}{2}\right) \phi(x - n) dx.$$

We can also write:

$$Q_1 f = \sum_n d_n^1 \psi_{1n}.$$

Then  $d_n^1 = \langle Q_1 f, \psi_{1n} \rangle = \sum_n c_n^0 \langle \phi_{0n}, \psi_{1n} \rangle = \sum_n c_n^0 g_{n-2k}$ , where

$$g_n = 2^{-1/2} \int \psi\left(\frac{x}{2}\right) \phi(x - n) dx.$$

As shown in Heijmans [3], it is possible to choose  $\psi$  in such a way that  $g_n = (-)^n \overline{h_{1-n}}$ . As we will choose  $\phi$  to be real valued, this implies  $g_n = (-)^n h_{1-n}$ .

By repeating this procedure  $N$  times we arrive at the decomposition

$$f = Q_1 f + Q_2 f + \dots + Q_N f + P_N f$$

where

$$P_n f = \sum_k c_k^n \phi_{nk}$$

and

$$Q_n f = \sum_k d_k^n \psi_{nk}.$$

The coefficients  $c_k^n$  and  $d_k^n$  can be calculated from the recursion

$$\begin{aligned} c^j &= \mathbf{H}c^{j-1} \\ d^j &= \mathbf{G}c^{j-1} \end{aligned}$$

where

$$\begin{aligned} (\mathbf{H}a)_k &= \sum_n h_{n-2k} a_n, \\ (\mathbf{G}a)_k &= \sum_n g_{n-2k} a_n. \end{aligned}$$

These operations can be performed easily and are suitable for implementation in hardware. After a number of iterations, the original sequence  $c^0$  is decomposed into a lowest resolution signal  $c^N$  and difference signals  $d^N, d^{N-1}, \dots, d^1$  of finer and finer resolution. The original image can be reconstructed by repeated use of the relation

$$P_{j-1} f = P_j f + Q_j f = \sum_n c_k^j \phi_{jk} + \sum_n d_k^j \psi_{jk}.$$

This implies

$$\begin{aligned} c_n^{j-1} &= \langle P_{j-1} f, \phi_{j-1,n} \rangle \\ &= \sum_k c_k^j \langle \phi_{jk}, \phi_{j-1,n} \rangle + \sum_k d_k^j \langle \psi_{jk}, \phi_{j-1,n} \rangle \\ &= \sum_k c_k^j h_{n-2k} + \sum_k d_k^j g_{n-2k}. \end{aligned}$$

The reconstruction algorithm can therefore be described by the recursion

$$c^{j-1} = \mathbf{H}^* c^j + \mathbf{G}^* d^j$$

where

$$\begin{aligned} (\mathbf{H}^* a)_n &= \sum_k h_{n-2k} a_k, \\ (\mathbf{G}^* a)_n &= \sum_k g_{n-2k} a_k. \end{aligned}$$

The result of the recursion is

$$c^0 = \sum_{j=0}^N (\mathbf{H}^*)^{j-1} \mathbf{G}^* d^j + (\mathbf{H}^*)^N c^N.$$

A two-dimensional multiresolution representation can be reconstructed using products of the functions defining the one dimensional multiresolution representation. The linear space of functions defined on  $\mathbb{Z}^2$  is the tensor product of two spaces of functions on  $\mathbb{Z}$ :

$$V_i = V_i^{(x)} \otimes V_i^{(y)}.$$

The one dimensional decomposition can be used to construct a two dimensional decomposition:

$$\begin{aligned} V_i^{(x)} \otimes V_i^{(y)} &= (V_{i+1}^{(x)} \oplus W_{i+1}^{(x)}) \otimes (V_{i+1}^{(y)} \oplus W_{i+1}^{(y)}) = \\ &= (V_{i+1}^{(x)} \otimes V_{i+1}^{(y)}) \oplus (V_{i+1}^{(x)} \otimes W_{i+1}^{(y)}) \oplus (W_{i+1}^{(x)} \otimes V_{i+1}^{(y)}) \oplus (W_{i+1}^{(x)} \otimes W_{i+1}^{(y)}). \end{aligned}$$

The different resolution spaces are :

$$V_N = \text{span}\{\phi_{Nn}(x)\phi_{Nm}(y) | m, n \in \mathbb{Z}\}.$$

The orthogonal complements are spanned by three types of functions:

$$W_N = \text{span}\{\phi_{Nn}(x)\psi_{Nm}(y), \psi_{Nn}(x)\phi_{Nm}(y), \psi_{Nn}(x)\psi_{Nm}(y) | m, n \in \mathbb{Z}\}.$$

The multiresolution decomposition is described by the recursions

$$\begin{aligned} P_j f &= \sum_{mn} c_{mn}^j \phi_{jm}(x) \phi_{jn}(y), \\ Q_j f &= \sum_{mn} \left\{ d_{mn}^{(x)j} \psi_{jm}(x) \phi_{jn}(y) + d_{mn}^{(y)j} \phi_{jm}(x) \psi_{jn}(y) + d_{mn}^{(xy)j} \psi_{jm}(x) \psi_{jn}(y) \right\}. \end{aligned}$$

The coefficients can be found by the recursion relations

$$\begin{aligned} c^{j+1} &= \mathbf{H}_x \mathbf{H}_y c^j, \\ d^{(x)j+1} &= \mathbf{G}_x \mathbf{H}_y c^j, \\ d^{(y)j+1} &= \mathbf{H}_x \mathbf{G}_y c^j, \\ d^{(xy)j+1} &= \mathbf{G}_x \mathbf{G}_y c^j. \end{aligned}$$

The operators  $\mathbf{H}_x$ ,  $\mathbf{G}_x$  and  $\mathbf{H}_y$ ,  $\mathbf{G}_y$  act on the first and second indices, respectively. One has for example:

$$(\mathbf{H}_x a)_{pq} = \sum_n h_{n-2p} a_{nq}.$$

In two dimensions, the size of a low scale image is a quarter of the size of the original image. Hence the number of coefficients needed to describe the difference signal is three times as large as the number of coefficients needed to describe the low

resolution image. The signal  $c^{j+1}$  is the coarse scale information. There are three different signals:  $d^{(x)j+1}$  indicates small scale variations where the signal varies in the x-direction, but not in the y-direction. Thus a high value of  $d^{(x)j+1}$  indicates the presence of a vertical edge in the image. Likewise, a large value of  $d^{(y)j+1}$  indicates the presence of a horizontal edge and a large value of  $d^{(xy)j+1}$  indicates the presence of a corner point.

The reconstruction algorithm is defined by the recursion

$$c^{j-1} = \mathbf{H}_x^* \mathbf{H}_y^* c^j + \mathbf{G}_x^* \mathbf{H}_y^* d^{(x)j} + \mathbf{H}_x^* \mathbf{G}_y^* d^{(y)j} + \mathbf{G}_x^* \mathbf{G}_y^* d^{(xy)j}.$$

Images are defined on a regular grid. Most often the grid is a square, the length of the edge being a power of 2. The initial sequence  $c^0$  therefore has a finite number of nonzero entries. In order to be able to perform the calculations, one must choose  $\phi$  in such a way that only a finite number of the coefficients  $h_n$  have a nonzero value. If such a choice is made, the total number of nonzero coefficients in the sequences  $c^{j+1}$ ,  $d^{(x)j+1}$ ,  $d^{(y)j+1}$ , and  $d^{(xy)j+1}$  is only slightly larger than the number of nonzero coefficients in the sequence  $c^j$ . The increase is caused by edge effects. If it is assumed that the image is periodic across the boundaries, the edge effects are eliminated and there is no increase in the number of coefficients to be stored after each iteration.

The multiresolution decomposition described above depends on the choice of an appropriate function  $\phi$ . Yet the algorithm uses only the coefficients  $h_n$  and  $g_n$ . Daubechies has derived conditions that the  $h_n$  and  $g_n$  must satisfy in order to make the algorithm work, without any reference to the function  $\phi$ . These conditions are:

$$\begin{aligned} \sum_n |h_n| < \infty, \quad \sum_n |g_n| < \infty; \\ \sum_n \{h_{n-2k} h_{n-2l} + g_{n-2k} g_{n-2l}\} &= \delta_{kl}; \\ \sum_n h_{n-2k} g_{n-2l} &= 0; \\ \sum_n h_n &= \sqrt{2}, \quad \sum_n g_n = 0. \end{aligned}$$

The second and third conditions guarantee that the original image can be reconstructed from the decomposition and that the decomposition contains no redundant information. The fourth conditions indicates that  $H$  acts as an averaging operator, while  $G$  acts like a local detail filter.

### §3 Compression rate and reconstruction errors

It has been argued before that the possibility of image compression is due to the predictability of natural images, or to be more precise, the fact that spatial correlations between pixel values exist. In order to achieve sufficient compression, the low level resolution functions must be a good prediction for the true image. This means that images which contain only energy at the higher levels must be sufficiently regular. Remember that (in one dimension) the reconstruction equation reads:

$$c^0 = \sum_{j=0}^N (\mathbf{H}^*)^{j-1} \mathbf{G}^* d^j + (\mathbf{H}^*)^N c^N.$$

We are therefore interested in sequences of the form  $(\mathbf{H}^*)^N e$  where  $e$  has only one nonzero entry. These sequences must be sufficiently regular for high values of  $N$ .

In stead of looking at the sequences themselves, we will study the functions defined by

$$\eta_l(x) = (T_H^l \chi_{[-1/2, 1/2]})(x),$$

where  $\chi$  is the indicator function and the operator  $T_H$  is defined by

$$(T_H f)(x) = \sqrt{2} \sum_n h_n f(2x - n).$$

Thus problems with the spacing between the grid points at different resolution are avoided. Moreover, this way of looking at things makes it possible to use the concept of continuity of functions. By taking the Fourier transform one gets:

$$\hat{\eta}_l(\xi) = \left[ \prod_{j=1}^l H(2^{-j}\xi) \right] \frac{\sin(2^{-l-1}\xi)}{2^{-l-1}\xi}$$

where  $H(\xi) = 2^{-1/2} \sum_n h_n e^{in\xi}$ . Daubechies has proven the following

**Theorem 3.1.** *Let  $h_n$ ,  $H(\xi)$  and  $\eta_l$  be defined as described above. Suppose that  $H(\xi)$  has the form*

$$H(\xi) = \left[ \frac{1}{2}(1 + e^{i\xi}) \right]^N \sum_n f_n e^{in\xi}$$

*such that  $\sum_n |f_n| |n|^\epsilon < \infty$  for some  $\epsilon > 0$  and  $\sup |\sum_n f_n e^{in\xi}| = B < 2^{N-1}$ . Then  $\eta_l$  converges pointwise to the continuous function  $\eta_\infty$  defined by*

$$\hat{\eta}_\infty(\xi) = \prod_{j=1}^{\infty} H(2^{-j}\xi).$$



It can be shown that

$$\left| \prod_{j=1}^{\infty} H(2^{-j}\xi) \right| = \mathcal{O}((1 + |\xi|)^{N - \log B}).$$

This implies that the regularity of  $\eta_{\infty}$  can be increased if one succeeds in choosing  $h_n$  such that  $\log B - N$  is sufficiently large. Daubechies has been able to construct in this manner a family of wavelets of arbitrarily high regularity and with compact support.

Exact reconstruction of the image is only possible if the decomposition coefficients are known exactly. As the coefficients are not integers, this is not possible. In practice, a quantisation of the coefficients is necessary. As the wavelet basis is orthonormal, the root-mean-square (RMS) error in the reconstructed image can be expressed as

$$\epsilon = \left[ \epsilon_N^2 + \sum_{k=1}^N [(\epsilon_k^x)^2 + (\epsilon_k^y)^2 + (\epsilon_k^{xy})^2] \right]^{1/2}.$$

Here  $\epsilon_k^x$ ,  $\epsilon_k^y$ , and  $\epsilon_k^{xy}$  are the errors in the difference signals and the  $\epsilon_N$  is the error in the lowest resolution image ( $\epsilon_N^2 = \sum (c_{ij}^N(\text{true}) - c_{ij}^N(\text{quantised}))^2$ ). The higher levels contain less coefficients than the lower levels, but the coefficients on the higher levels tend to be larger than those on lower levels. Therefore the different levels make a contribution to the overall error of comparable magnitude.

Quantisation is a subject by itself. It stretches too far to treat it in this chapter, but we will make a few remarks. The true coefficients are real numbers. Quantisation can be done by selecting a finite number of reals which form a so-called quantisation grid. Each coefficient is then replaced by the nearest point in the grid. The most straightforward way of choosing the grid is taking all grid points evenly separated. A more sophisticated choice is motivated by the fact that most coefficients in the detail signal have values near 0. Therefore, choosing the grid points not equally spaced, but closer together near 0 can reduce the total error. For some coefficients far away from 0, the error can become larger and this can cause typical artefacts in the reconstructed image.

A different approach to data reduction is to send only those coefficients whose value is large. Small coefficients contribute only very little to the signal and can be deleted without causing a large error.

We have implemented a quantisation scheme using grids with equally separated grid points. The grid point separation is the same on all levels of the decomposition. Even with such a simple quantisation scheme, bit rates as low as 0.4 bits per pixel can be achieved while the quality of the reconstruction is still sufficiently high. Due to the localisation of the low level basis functions, small details in the image, like small bright or dark spots, are preserved better than in most other image compression schemes.

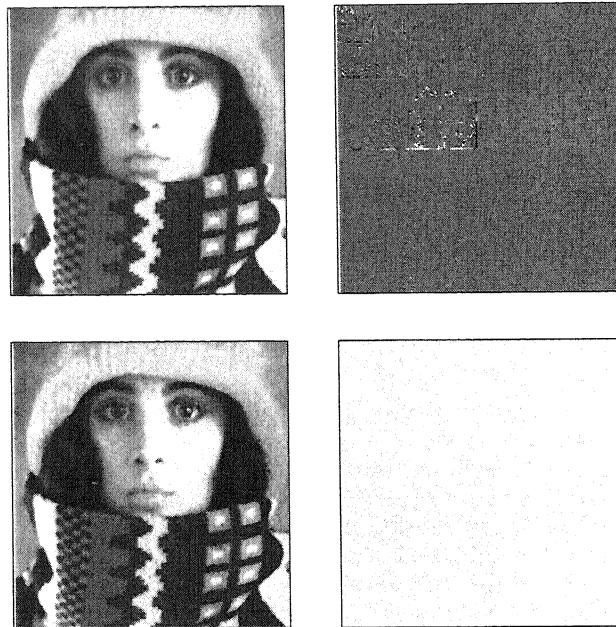


Figure 1. Results from the compression method.

The pictures show an example. The original image is shown in the upper left hand corner. The upper right hand corner represents the coefficients of the decomposition. The lower left hand quarter, the upper right hand corner and the lower right hand corner of the image show the coefficients  $d^{(x)ij}$ ,  $d^{(y)ij}$ , and  $d^{(xy)ij}$ , respectively. The upper left hand corner shows the decomposition of the low resolution image. This low resolution image is decomposed in the same way as the original image and this procedure is repeated recursively, giving a nested decomposition. The lower left hand image shows the reconstructed version. The lower right hand image shows the difference between the original image and the reconstruction. This image is rescaled for display purposes. In this example, a data reduction of 95% (0.4 bpp) has been achieved.

References

1. Burt, P. and E. Adelson, The laplacian image pyramid as a compact image code, *IEEE Trans. Comm.* **31** (1983), 482–540.
2. Daubechies, I., Orthonormal bases of compactly supported wavelets, *Comm. Pure and Appl. Math.* **41** (1988), 909–996.
3. Heijmans, H., Discrete wavelets and multiresolution analysis, this volume.
4. Mallat, S., A theory for multiresolution signal decomposition: the wavelet representation, *IEEE Trans. Pattern Anal. Machine Intell.* **11** (1989), 674–692.
5. Shannon, C.E. and W. Weaver, *The Mathematical Theory of Computation*, University of Illinois Press, Illinois, 1949.

*Peter Nacken*  
CWI  
P.O. Box 4079  
1009 AB Amsterdam  
The Netherlands  
nacken@cwi.nl