Appendix 3. Wavelet compression

1 Introduction

The DFT and DCT are linear transforms based on decomposition of an input signal over a system of orthogonal harmonic functions. The main shortcoming of these transforms is that basis functions are uniformly distributed over the frequency axis. It means that all frequencies of the input signal are considered as equally important in terms of recovering the original signal from the set of transform coefficients. On the other hand, from the signal reconstruction point of view it is more important to preserve the high quality of low-frequency components of the signal than to preserve its high-frequency components. Thus, the resolution of the system of basis functions should be non-uniform over the frequency axis. The problem of constructing such a transform is solved by using filter banks. One of the most efficient transforms is based on wavelet filter banks and called wavelet filtering.

The wavelet filter banks have special properties. The most important feature of these filter banks is their hierarchical structure. The input signal is decomposed using two filters into low-frequency and high-frequency parts. Then each component of the input signal is decimated, that is, the only even-numbered samples are kept. The downsampled high-frequency part represents a final output because it is not transformed again. Since this part of the signal contains a rather insignificant part of the signal energy it can be encoded by using a small number of bits. The decimated low-frequency component contains the main part of the signal energy and it is filtered again by the same pair of filters. The decimated high-frequency part of the low-frequency signal component is not transformed again but the decimated low-frequency part of the low-frequency signal component can be filtered again and so on. By choosing a filter bank in a proper way it is possible to provide larger compression ratio compared to DCT based codecs on the assumption of the same quality of the synthesized signal.
Thus, the main idea of the wavelet transform is a hierarchical decomposition of the input sequence into the so-called \textit{reference} (low-frequency) subsequences with diminishing resolutions and related with them the so-called \textit{detail} (high-frequency) subsequences. At each level of decomposition, the wavelet transform is invertible, that is, the reference signal of this level together with the corresponding detail signal provides perfect reconstruction of the reference signal of the next level (with higher resolution).

Fig. 1 illustrates one level of wavelet decomposition followed by a reconstruction. The input sequence $x(n)$ is filtered by a lowpass filter with the pulse response $h_0(n)$ and by a highpass filter with the pulse response $h_1(n)$. The downsampling step is symbolized by (\downarrow 2). The sequence $r_1(n)$ is the reference signal (decimated result of lowpass filtering), and $d_1(n)$ is the detail signal (decimated result of highpass filtering). It is evident that this scheme transforms one sequence of length $N$ into two subsequences of length $N/2$ each.

![Figure 1: Wavelet decomposition](image)

In the theory of wavelet filter banks such pairs of filters $h_0(n)$ and $h_1(n)$ are found that there exist pairs of the inverse filters $g_0(n)$ and $g_1(n)$ providing the perfect reconstruction of the input signal. To reconstruct the input signal from signals $r_1(n)$ and $d_1(n)$ these signals are first upsampled with a factor of 2. In Fig. 1 upsampling is symbolized by (\uparrow 2). Then the upsampled low-frequency and high-frequency components are filtered by the inverse lowpass filter with the pulse response $g_0(n)$ and the inverse highpass filter with the pulse response $g_1(n)$, respectively. The sum of the results of filtering is the output signal $y(n)$. The wavelet transform (wavelet filtering) provides perfect reconstruction of the input signal, that is, the output signal is determined as

$$y(n) = Ax(n - \alpha)$$

where $A$ is the gain factor, and $\alpha$ is the delay.
In the case of multilevel decomposition the reference signal \( r_1(n) \) represents the input signal of the next decomposition level. Filtering is performed iteratively as shown in Fig. 2.

\[
\begin{align*}
\mathbf{x}(n) & \xrightarrow{h_0(n)} r_1(n) \xrightarrow{\downarrow 2} r_2(n) \\
\mathbf{x}(n) & \xrightarrow{h_1(n)} d_1(n) \xrightarrow{\downarrow 2} d_2(n)
\end{align*}
\]

Figure 2: Multiresolution wavelet decomposition

At the \( L \)th level of decomposition we obtain the reference signal \( r_L(n) \) with resolution \( 2^L \) times scaled down compared to the resolution of the input signal and the detail signals \( d_L(n), d_{L-1}(n), \ldots, d_1(n) \) with resolution \( 2^j, j = L, L-1, \ldots, 1 \) times scaled down compared to the input signal, respectively. Each detail signal \( d_i(n) \) contains such information that together with the reference signal \( r_i(n) \) it allows recovering of \( r_{i-1}(n) \) which represents the reference signal of the next level. At the \( L \)th level of the decomposition, the total length of reference and detail subsequences is

\[
2^{-L}N + 2^{-L}N + 2^{-(L-1)}N + 2^{-(L-2)}N + \ldots + 2^{-1}N
\]

\[
= N \left( \sum_{i=1}^{L} 2^{-i} + 2^{-L} \right)
\]

\[
= N \left( 2^{-1} \frac{1 - 2^{-L}}{1 - 2^{-1}} + 2^{-L} \right) = N.
\]

Consider how wavelet filtering can be used in order to perform the \( r \)-level wavelet decomposition of the image of the size \( M \times N \) pixels (to be more precise, usually we decompose one of the so-called \( Y \), \( U \) or \( V \) components of the original image or a matrix of size \( M \times N \)). It is evident that the two-dimensional wavelet decomposition is a separable transform. Thus, first we perform the wavelet transform over the matrix rows and then the obtained matrix is filtered over the columns. At the first level of the wavelet hierarchical decomposition, the image is decomposed using two times downsampling (over the rows and over the columns) into high horizontal-high
vertical (HH1), high horizontal-low vertical (HL1), low horizontal-high vertical (LH1), and low horizontal-low vertical (LL1), frequency subbands. They correspond to filtering by highpass filter $h_1(n)$ over rows and over columns, by highpass filter $h_1(n)$ over rows and lowpass filter $h_0(n)$ over columns, by lowpass filter $h_0(n)$ over rows and highpass filter $h_1(n)$ over columns and by lowpass filter $h_0(n)$ over rows and columns, respectively. The LL1 subband is then further filtered and downsampled two times to produce a set of HH2, HL2, LH2 and LL2 subbands.

Figure 3: Wavelet decomposition of image

This is done recursively $r$ times to produce an array such as that illustrated in Fig. 3, where filtering and downsampling (over the rows and over the columns each) were carried out three times. As the result we obtain $3r + 1$ matrices of reducing size. Most of the energy is in the low-lowpass subband LL3. This upper left subimage is a coarse approximation of the original image.

Each matrix is quantized by a scalar or vector quantizer and then encoded. The quantization step is chosen depending on the required compression ratio and bit allocation. Clearly, we can roughly quantize the subbands with low
energy without a significant loss of the reconstructed image quality.

The quantized highpass subbands usually contain many zeros. They can be efficiently lossless encoded using zero run-length coding followed by the Huffman coding of pairs (run length, amplitude) or arithmetic coding. The lowpass subbands usually do not contain zeros at all or contain small number of zeros and can be coded by the Huffman code or by the arithmetic code.

More advanced coding procedures used, for example, in the MPEG-4 standard try to take into account dependencies between subbands. One such method is called zero-tree coding. Fig. 4 illustrates the parent-child dependencies of subbands. A single parent node in a subband of higher decomposition level has four children nodes corresponding to the in the corresponding subband of lower decomposition level. Each child node has four corresponding next generation children nodes in the same type subband of the next level.

Figure 4: The wavelet hierarchical subband decomposition and the parent-child dependencies of subbands

Fig. 5 is a flowchart for encoding a coefficient of the significance map or, in other words, this picture illustrates how the zerotree-based encoder classifies
the wavelet coefficients and generates a zerotree. One of approaches typically used to lossless encode wavelet subband coefficients consists of encoding the so-called *significance map*, i.e., the binary decision whether a coefficient has zero or nonzero quantized value followed by encoding coefficient magnitudes. Rather large fraction of the bit budget is usually spent for encoding the significance map. In order to reduce the number of bits for significance map coding, the zerotree method implies the following classification of wavelet coefficients. A coefficient is said to be an element of a zerotree for the given threshold if itself and all of its descendants (children) are insignificant with respect to this threshold. An element of a zerotree is a *zerotree root* if it is not the descendant of a previously found zerotree root, that is, it is not predictably insignificant from a zerotree root at a coarser scale. A zerotree root is encoded with a special symbol indicating that all its descendants at the finer scales are insignificant. Thus, the following four symbol types are used: zerotree root, isolated zero, which means that the coefficient is insignificant but has some significant descendants, positive significant symbol, and negative significant symbol.

Most often used filters for wavelet decomposition are

\[
\begin{align*}
  h_0(n) &= (-0.176777, 0.353553, 1.060660, 0.35353, -0.176777), \\
  h_1(n) &= (-0.353553, 0.707107, -0.35353).
\end{align*}
\]

which can be also given in the form:

\[
\begin{align*}
  h_0(n) &= (-1, 2, 6, 2, -1)\sqrt{32}, \\
  h_1(n) &= (-2, 4, -2)/\sqrt{32}.
\end{align*}
\]

Corresponding reconstructing filters are

\[
\begin{align*}
  g_0(n) &= (0.353553, 0.707107, 0.353553), \\
  g_1(n) &= (-0.176777, -0.353553, 1.060660, -0.35353, -0.176777)
\end{align*}
\]

or in other form:

\[
\begin{align*}
  g_0(n) &= (2, 4, 2)/\sqrt{32}, \\
  g_1(n) &= (-1, -2, 6, -2, -1)/\sqrt{32}.
\end{align*}
\]
Input coefficient

Is coefficient significant? NO

YES

What sign? (+) (-)

Does coefficient descend from a zerotree root?

YES

Predictably insignificant
Don’t code

NO

Does coefficient have significant descendants?

YES

NOP 0

NO

Code positive symbol

Code negative symbol

Code isolated zero symbol

Code zerotree root symbol

Figure 5: Zerotree coding