IV Signal compression

May 31, 2009

1 Introduction

The goal of signal compression is to reduce the storage or transmission requirements of a signal. A scheme which achieves this goal is called a *transform coding scheme.*

A transform coding scheme consists of 3 steps:

1. The transform step.
2. The quantization step.
3. The coding step.

The purpose of the transform step is to decorrelate the data by removing redundancy. This is acheived mainly by replacing the signal with a sequence of transform coeffieicnts. This step is usually done with an invertible transform (Fourier Transform, Gabor Transform, Wavelet Transform, ... etc). Since the transform is invertible, this step is lossless. No information is lost.

The quantization step reduces the typically large set of transform coefficients by a representative finite set of integer values. This process is not invertible and therefore, it is lossy.

The coding step takes advantage of the large number of zeros produced in the transform step and the quantization step to produce a compressed coded image.

2 The transform step

When a signal contains large regions of constant or smoothly changing values, its wavelet transform with wavelets that have a sufficient number of vanishing moments produce very small coefficients. In the next section we will see how this can be used to achieve a high compression ratio.

For a given signal f, it is possible to choose a "best" wavelet basis, e.g., one which maximizes the number of coefficients below a prespecified threshold. The disadvantage, of course, is the added overhead required to specify information about the wavelet basis. This disadvantage can be greatly reduced by specifying a best basis for a group of signals rather than one for each signal. This is the case, for example, with fingerprinting. The ridges on a typical fingerprint translate to a rapid oscillation in pixel values; thus, it is not surprising that standard wavelet basis does not give the optimal representation.

Factors governing the choice of a wavelet basis include the following considerations.

a. **Symmetry:** Many signals in applications are symmetric, e.g., periodic signals and images. The orthogonal wavelet trasfrom of these signals produce artificial oscillations due to sudden jumps. The presence of these oscillations means large coefficients which compromise the goal of compressing signals by producing as many small coefficients as possible. Symmetric filters are desirable because they eliminate the artificial oscillations. Since orthogonal wavelets cannot be symmetric, biorthogonal wavelets are used, whenever appropriate, with signal processing.

b. **Vanishing moments:** A large number of vanishing moments results via smoothness of the wavelet in small coefficients on the analysis side and less blocking artifacts on the reconstruction side. Therefore, it is desirable to have wavelets with a large number of vanishing moments.

c. **Filter size:** Although long filters mean smooth wavelets and hence, good compression properties, such filters tend to be oscillating. Such oscillations can be visible in the reconstructed signal. Therefore, we seek filters with the shortest possible length. The 9/7 biorthogonal wavelets provide a good compromise between the number of vanishing moments and the length of the filter. In fact these are the wavelets used for fingerprint compression.

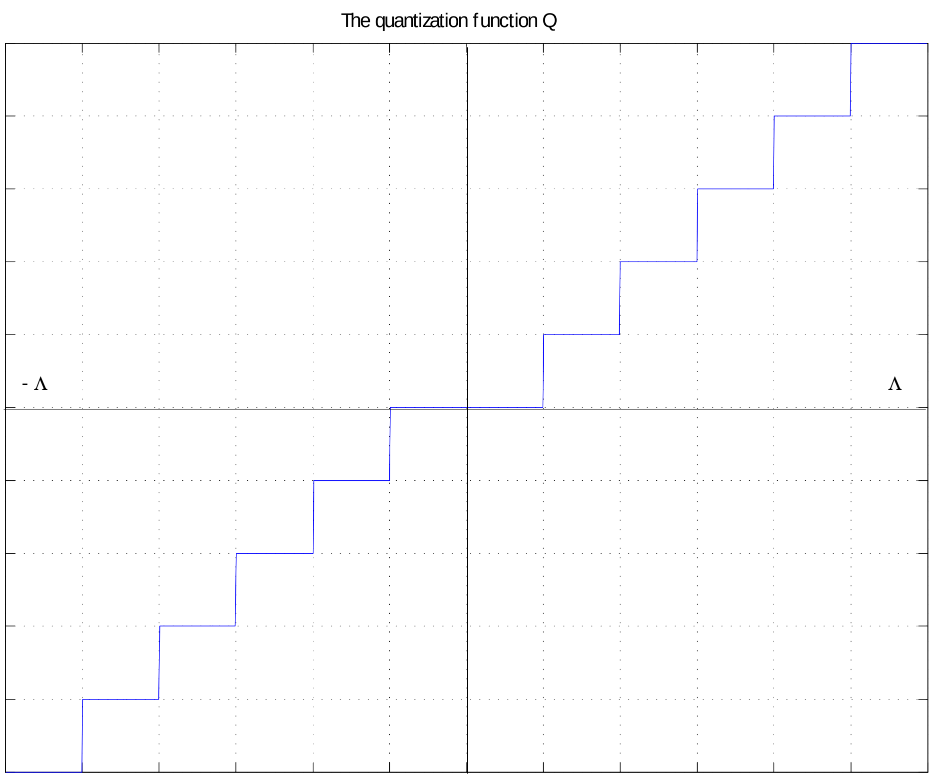
3 Thresholding and the quantization step

The trasform step replaces the signal f with a sequence F of M (possibly) high-precision floating-point coefficients. Quantization reduces the number of values that these coefficients take by representing this set of coefficients by means of a finite set of integers. One way to implement such a quantization is called *scalar quantization*. The procedure is as follows.

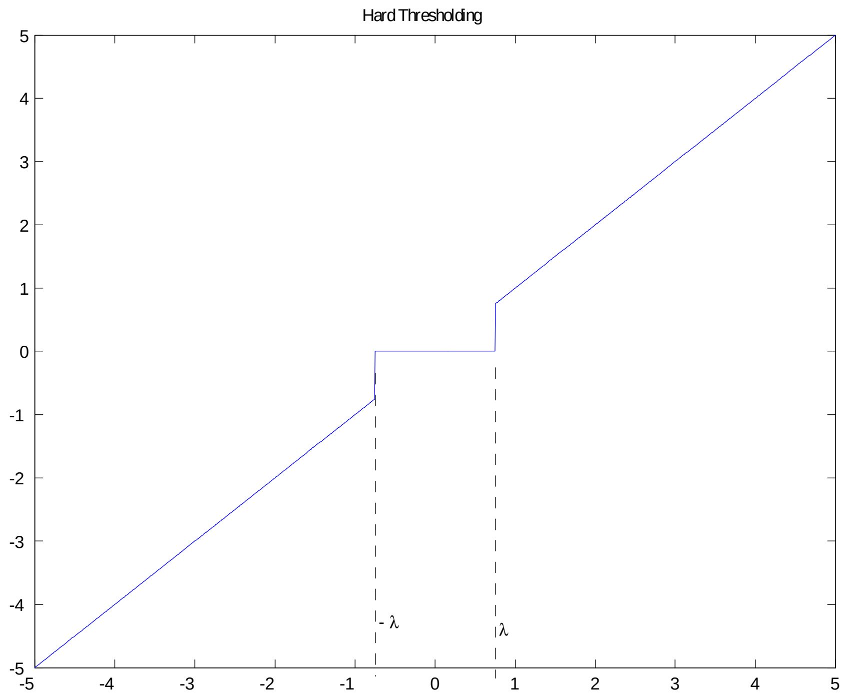
1. Suppose all coefficients are in the range $[-\Lambda, \Lambda]$, that is, $F_k \in [-\Lambda, \Lambda]$, and they are to be quantized by a number of $2q$ levels. Partition the interval $[-\Lambda, \Lambda]$ into $2q$ equal intervals $[x_{-q+1}, x_{-q+2})$, $[x_{-q+2}, x_{-q+3})$, $\cdots, [x_{q-1}, x_q]$, where $x_{-q+1} = -\Lambda,\ x_{i+1} - x_i = 2\Lambda/q$.
2. Define a quantization map $Q : [-\Lambda, \Lambda] \to \{-q+1, -q+2, \cdots, q\}$ by

$$Q(x) = \begin{cases} j & \text{if } x \in [x_j, x_{j+1}) \text{ and } x_{j+1} \leq 0, \\ j+1 & \text{if } x \in [x_j, x_{j+1}) \text{ and } x_j \geq 0 \end{cases} .$$

The graph of this quantization map is shown below.



Observe that the quantization function Q maps the double-wide interval $[x_{-1}, x_1) = \left[-\frac{\Lambda}{q}, \frac{\Lambda}{q}\right)$ to zero. If the signal is smooth, this interval contains most of the wavelet coefficients. In this respect, the quantization process works as a threshold. More generally, a threshold parameter $\lambda > 0$ is set such that all coefficients of absolute value less than λ is set to zero before applying the quantization. There are



two types of thresholding: hard thresholding and soft thresholding. Hard thresholding is implemented as follows

$$T_{\text{hard}}(x) = \begin{cases} 0 & \text{if } |x| \le \lambda \\ x & \text{if } |x| > \lambda \end{cases}.$$

Soft thresholding is implemented as follows

$$T_{\text{soft}}(x) = \begin{cases} 0 & \text{if } |x| \le \lambda \\ x - \lambda & \text{if } x > \lambda \\ x + \lambda & \text{if } x < -\lambda \end{cases}.$$

The combined effect of applying thresholding and quantization is the composition $Q \circ T$.

3.1 The coding step

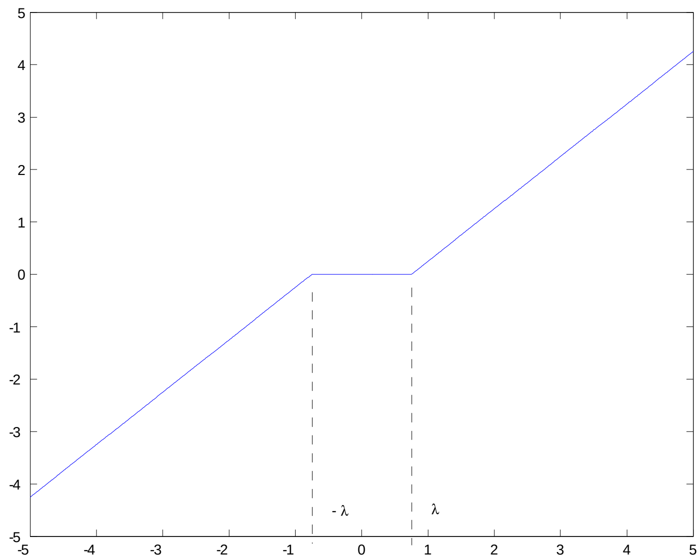
At this stage a signal of length M is transformed and quantized as a string of M integers in the range $\{-q+1, -q+2, \cdots, q\}$. Coding consists of assigning binary values to each integer in the quantized signal for storing or transmission purposes. We make use of the relative frequencies with which these values occur in order to reduce the number of bits required to represent the signal.

Example 1 *Suppose $q = 2$ and $M = 16$. Then a signal is a string of the form*

$$AABCDAAABBADAAAA,$$

where we replaced the integers $-1 : 2$ with the symbols $A - D$. A possible coding scheme is

$$\begin{aligned} A &\rightarrow 00 \\ B &\rightarrow 01 \\ C &\rightarrow 10 \\ D &\rightarrow 11 \end{aligned}$$



Accordingly, the coded image is

$$00000110111000000010100110000000,$$

a total of 32 bits. On the other hand, observing that the symbol A appears far more frequently than the other symbols, we may take advantage of this by assigning A the shortest code word, namely 0. We can repeatedly do this and devise the coding scheme

$$\begin{array}{ccc} A & \rightarrow & 0 \\ B & \rightarrow & 10 \\ D & \rightarrow & 110 \\ C & \rightarrow & 111 \end{array}$$

According to this coding scheme, the image is coded as

$$0010111110000101001100000,$$

a total of 25 bits and a saving of about 22%.

3.2 Sources and codes

In this subsection we present some basic concepts of information and coding theory and introduce the concept of entropy of a symbolic source.

Definition 2 *(symbol sources, codewords, binary codes, coding schemes and average codeword lenght)*

(i) *A symbol source is a pair (S,P) where S is finite sequence $S=\{s_1,s_2,\cdots,s_q\}$ and P is a corresponding sequence of assigned probabilities $P=\{p_1,p_2,\cdots,p_q\}$, where $p_i=P(s_i)$, $0\le p_i\le 1$, $1\le i\le q$,and $\sum_{i=1}^q p_i=1$.*

(ii) *A codeword, w, is a finite string of binary digits:*

$$w=d_1d_2\dots d_k,\ \ \textit{where}\ d_j\in\{0,1\},\ 1\le j\le k.$$

(iii) *A binary code, C, is a finite sequence of codewords:*

$$C=\{w_1,w_2,\dots,w_p\}.$$

(iv) *A coding scheme is a one to one mapping* $\sigma : S \to C$.

(v) *The average codeword length* $\mathrm{ACL}(\sigma)$ *of a coding scheme* σ *is defined as*

$$\mathrm{ACL}(\sigma) = \sum_{i=1}^{q} p_i \ell(w_i),$$

where $w_i = \sigma(s_i), 1 \leq i \leq q$ *and* $\ell(w)$ *is the length of the codeword* w.

Example 3 *In the previous example, the symbol source* (S, P) *is* $S = \{A, B, C, D\}$ *and* $P = \left\{\frac{5}{8}, \frac{3}{16}, \frac{1}{16}, \frac{1}{8}\right\}$, *where* $P(A) = \frac{5}{8},\ P(B) = \frac{3}{16},\ P(C) = \frac{1}{16},\ P(D) = \frac{1}{8}$. *The two coding schemes discussed above are*

1. *The codewords are* $w_1 = 00$, $w_2 = 01$, $w_3 = 10$ *and* $w_4 = 11$, *the binary code* $C = \{00, 01, 10, 11\}$ *and the coding scheme* σ *is* $\sigma(A) = 00,\ \sigma(B) = 01,\ \sigma(C) = 10, \sigma(D) = 11$. *For this scheme*

$$\mathrm{ACL}(\sigma) = \frac{5}{8} \cdot 2 + \frac{3}{16} \cdot 2 + \frac{1}{16} \cdot 2 + \frac{1}{8} \cdot 2 = 2,$$

which is not surprising since each codeword has length 2.

2. *The codewords are* $w_1 = 0$, $w_2 = 10$, $w_3 = 11$ *and* $w_4 = 111$, *the binary code* $C = \{0, 10, 11, 111\}$ *and the coding scheme* σ *is* $\sigma(A) = 0,\ \sigma(B) = 10,\ \sigma(C) = 111,\ \sigma(D) = 110$.*For this scheme*

$$\mathrm{ACL}(\sigma) = \frac{5}{8} \cdot 1 + \frac{3}{16} \cdot 2 + \frac{1}{16} \cdot 3 + \frac{1}{8} \cdot 3 = 1.562\,52.$$

The average codeword for this scheme is about 78% *as long as the less efficient scheme 1.*

In order that a coding scheme be uniquely deciphered, we require that no codeword should be the prefix of another codeword. (Consider, for example, the binary code in part 2 of the previous example.)

3.3 Entropy and information

The entropy $H(S)$ of a symbol source (S, P) is defined to be

$$H(S) = -\sum_{i=1}^{q} p_i \log_2(p_i).$$

It measures the degree of uncertainty of information associated with the source. This is because the function $x \log_2(x) = 0$ at $x = 0$ and $x = 1$. This means that any symbol with probability of occurring 0 does not add anything to the uncertainty of the information content of the source. The same holds true if a symbol has probability of occurring 1 (there can be at most one, the rest having probability zero). In this case, $H(S) = 0$. A simple exercise with Lagrange multipliers will show that $H(S)$ is maximum when each symbol is equally likely to appear. The common probability is $p_i = \frac{1}{q}$ and $H(S) = \log_2(q)$. In this case, if one symbol is put out by the source, it is still uncertain which symbol is to be put out next. The entropy function has the following property.

Lemma 4 *If two symbol sources* $A = \{a_1, a_2, \cdots, a_q\}$ *and* $B = \{b_1, b_2, \cdots, b_r\}$ *are independent then*

$$H(AB) = H(A) + H(B),$$

where AB *is the source defined by*

$$AB = \{a_i b_j\}_{1 \leq i \leq q, 1 \leq j \leq r}.$$

Proof. Since the sources A and B are independent, then $P(a_i b_j) = p_i p_j$. Hence,

$$\begin{aligned} H(AB) &= -\sum_{i=1}^{q}\sum_{j=1}^{r} p_i p_j \log_2(p_i p_j) \\ &= -\sum_{i=1}^{q} p_i \sum_{j=1}^{r} p_j (\log_2(p_i) + \log_2(p_j)) \\ &= -\sum_{i=1}^{q} p_i (\log_2(p_i) + H(B)) \\ &= -(H(A) + H(B)). \end{aligned}$$

■

3.4 Coding and compression

Given a symbol source $S = \{s_1, s_2, \cdots, s_q\}$ with assigned probabilities $p_i = P(s_i)$, suppose that $q = 2^s$. Then each symbol can be coded in s bits. If messages (signals, images, ... etc.) of length M are put out by the source, then each message can be stored in sM bits. A coding scheme σ tries to reduce this storage requirement by seeking alternative coding sets. For a given coding scheme σ, the average storage requirement of a message is $\mathrm{ACL}(\sigma) M$ bits. We then say that σ has a bit rate of $\mathrm{ACL}(\sigma)$. The *compression ratio* of σ is $s/\mathrm{ACL}(\sigma)$.

Theorem 5 *(best coding scheme)*
Let $m(S) = \min_\sigma \mathrm{ACL}(\sigma)$, then

$$H(S) \le m(S) \le H(S) + 1.$$

Example 6 *Suppose an image is quantized with $q = 32$ and that 95% of the transform coefficients are quantized to zero. Suppose further that the remaining 31 quantized values have the same probability of occurrence. Then*

$$P(0) = .95, \qquad P(i) = .05/31 \approx .0061,\ 1 \le i \le 31.$$

The entropy of this source is

$$H(S) = -.95\log_2(.95) - .05\log_2(.0061) \approx 0.534\,11.$$

Therefor, the best possible coding for this source has bit rate of $0.534\,11$ *and compression ratio of* $5/0.534\,11 \approx 9.4$.

The theoretical bit rate 0.53 in previous example is not achievable since any codeword of a coding scheme has length ≥ 1. Therefore, $\mathrm{ACL}(\sigma) \ge 1$. For example, the two schemes used in Example 3 have $\mathrm{ACL}(\sigma) = 2$, and 1.56, respectively. Thus there is a need to improve upon Theorem 5 to get as close as possible to the theoretical minimum. This is achieved by group coding.

Definition 7 *(symbol source extensions)*
Let $S = \{s_1, s_2, \cdots, s_q\}$ be a symbol source with associated probabilities $\{p_1, p_2, \ldots, p_q\}$. The n^{th} extension S^n of S is defined by

$$S^n = \{\{s_{i_1} s_{i_2}, \ldots, s_{i_n}\} : 1 \le i_1, i_2, \cdots, i_n \le q\}.$$

Observe that S^n is the set of all strings of length n of the symbols of S. There are q^n such strings. Since the symbols in S are independent, the associated probabilities for S^n are

$$\{p_{i_1} p_{i_2} \cdots p_{i_n}\}_{1 \le i_1, i_2, \cdots, i_n \le q}.$$

Theorem 8 $H(S^n) = nH(S)$.

Proof. This is a straightforward consequence of Lemma 4 and induction. ■

Theorem 9
$$H(S) \leq \frac{m(S^n)}{n} \leq H(S) + \frac{1}{n}.$$

Since any coding scheme for S^n can be regarded as a coding scheme for S, the above theorem means that by considering coding schemes for the n^{th} extensions of S we can find a coding scheme whose ACL is arbitrarily close to the theoretical minimum.

Example 10 *Let $S = \{A, B, C, D\}$ and consider the 64 symbol message*

$$AABCAAAAAAAAAAAAAAAABCAAAAAAAAAAAAAAAAAAAAAAAADA$$
$$AAAAAAAAACAAAAAA.$$

This message can be considered as a message composed of the symbols of

$$S^2 = \{AA, AB, AC, AD, BA, BB, BC, BD, CA, CB, CC, CD, DA, DB, DC, DD\}$$

The associated probabilities are

$$P(AA) = 23/32,\ P(BC) = 2/32,\ P(DA) = 1/32,\ P(AC) = 1/32,$$

the remaining probabilities being zeros. Using the coding scheme

$$\begin{array}{ccc} AA & \to & 0 \\ BC & \to & 10 \\ DA & \to & 110 \\ AC & \to & 111 \end{array}$$

the message is coded as

$$01000000000100000000000000110000011100 0,$$

a total of 38 bits. This is a $38/32 \approx 1.19$ bits per symbol if we consider the message to be of length 32 made up of the symbols of S^2, but a $38/64 \approx .59$ bits per symbol if we consider the message be of length 64 made up of the symbols of S. The theoretical optimal value in this case is 0.5857.

Although group coding produces bit rates closer to the optimal values, the price we pay is the overhead required to store or transmit the coding scheme. The size of S^n is q^n and if $q = 64$ and $n = 5$, then $64^5 > 1$ billion. This is a prohibitively high price.

3.5 The binary Huffman code

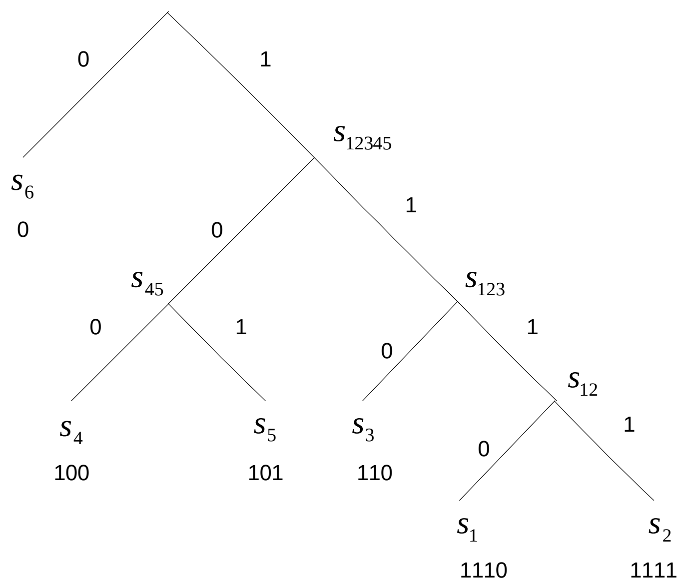
The binary Huffman code is a dynamical programming algorithm that constructs a coding scheme with minimum ACL for a given symbol source. It is defined recursively as follows.

Proposition 11 *(Huffman) Let $S = \{s_1, s_2, \cdots, s_q\}$ be a simple source with associated probabilities $\{p_1, p_2, \cdots, p_q\}$ where the symbols are arranged such that $p_1 \leq p_2 \leq \cdots \leq p_q$. Let $s_{12} = \{s_1, s_2\}$ and assign it the probability $p_{12} = p_1 + p_2$. An optimal coding scheme for S is obtained recursively by constructing an scheme for $S_1 = \{s_{12}, s_3, \cdots, s_q\}$ and then dividing the symbol s_{12} into its constituents s_1, s_2.*

Example 12 *We construct the optimal scheme for the 6 symbol source $S = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ with probabilities $\{0.05, 0.1, 0.1, 0.15, 0.2, 0.4\}$. Following is the sequence of recursive constructs together with the associated probabilities.*

$$\begin{aligned} S_1 &= \{s_3, s_{12}, s_4, s_5, s_6\}; \{0.1, 0.15, 0.15, 0.2, 0.4\} \\ S_2 &= \{s_4, s_5, s_{123}, s_6\}; \{0.15, 0.2, 0.25, 0.4\} \\ S_3 &= \{s_{123}, s_{45}, s_6\}; \{0.25, 0.35, 0.4\} \\ S_4 &= \{s_6, s_{12345}\}; \{0.4, 0.6\} \end{aligned}$$

The coding scheme is best explained by the following tree diagram. The codewords assigned to each symbol is shown below the symbol.



The Huffman optimal coding scheme for the 6-symbol source.

Therefore, we have the following coding scheme.

$$\begin{aligned} s_1 &\rightarrow 1110, \\ s_2 &\rightarrow 1111, \\ s_3 &\rightarrow 110, \\ s_4 &\rightarrow 100, \\ s_5 &\rightarrow 101, \\ s_6 &\rightarrow 0. \end{aligned}$$

The ACL for this scheme is

$$ACL(\sigma) = .05 \cdot 4 + 0.1 \cdot 4 + 0.1 \cdot 3 + 0.15 \cdot 3 + 0.2 \cdot 3 + 0.4 \cdot 1 = 2.35$$

The optimal bit rate for this source is

$$\begin{aligned} H(S) &= -(.05 \cdot \log_2(.05) + 2 \cdot 0.1 \cdot \log_2(.1) + 0.15 \cdot \log_2(.15) + 0.2 \cdot \log_2(.2) + 0.4 \cdot \log_2(.4)) \\ &= 2.2842. \end{aligned}$$

Let us compare the bit rate of the Huffman coding scheme to that of the coding scheme

$$\begin{aligned} s_1 &\rightarrow 11111, \\ s_2 &\rightarrow 1110, \\ s_3 &\rightarrow 1110, \\ s_4 &\rightarrow 110, \\ s_5 &\rightarrow 10, \\ s_6 &\rightarrow 0. \end{aligned}$$

This scheme has a bit rate of

$$ACL(\sigma) = .05 \cdot 5 + 0.1 \cdot 5 + 0.1 \cdot 4 + 0.15 \cdot 3 + 0.2 \cdot 2 + 0.4 \cdot 1 = 2.4$$

which is higher than the bit rate of the Huffman scheme.

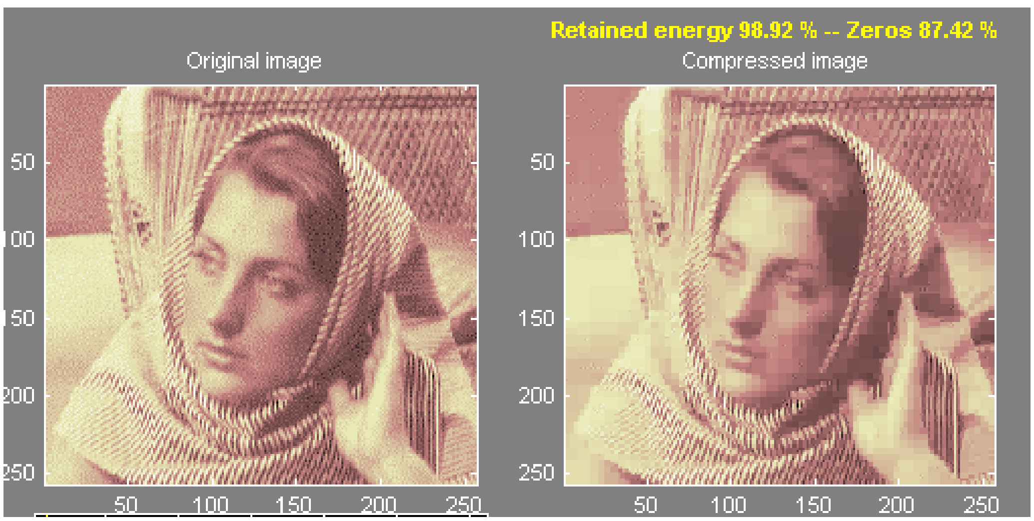
4 Image compression

A black and white image is an $M \times M$ array of integers in the range $[0, L-1]$. Each entry is called a picture element or a pixel. Each pixel value represents the gray scale level of that pixel, with 0 representing black and $L-1$ representing white. Usually, M is 256 or 512 and $L = 256$ (8 bits). Storing or transmitting an image requires $256 \times 256 \times 8 = 524\,288$ bits. The goal of image compression is to reduce the storage or transmission requirements of an image. A scheme which achieves this goal is called a *transform coding scheme*.

4.1 MATLAB examples

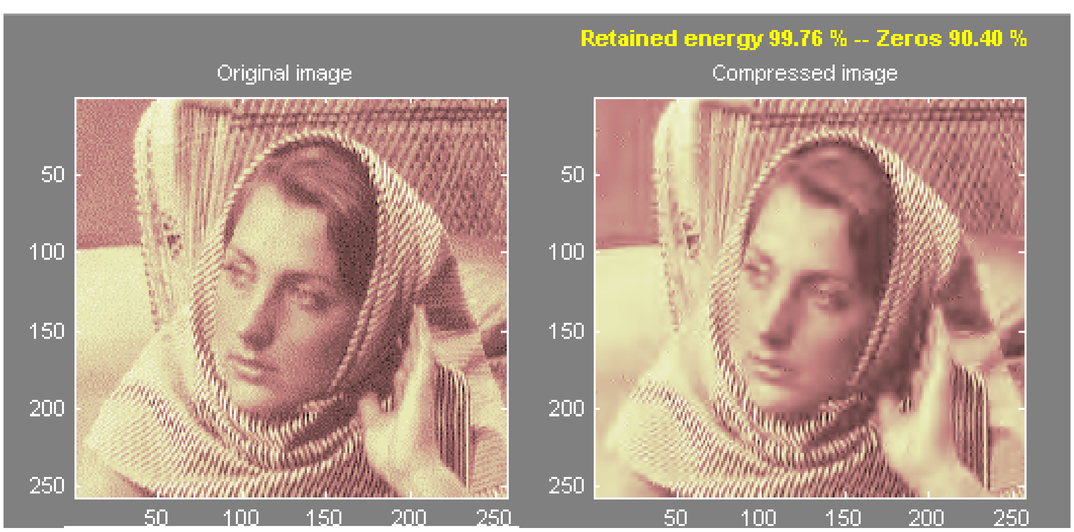
The MATLAB wavelet toolbox has the capability of compressing images. The compression is performed by taking the wavelet transform of an image, remove detail coefficients that are close to zero and then reconstruct the image. See the help documentation in MATLAB for more details.

The following image is analyzed with the Haar wavelet down to level 5 then compressed. 87.42% of the detail coefficients are set to zero. The reconstruction retains 98.92% of the original image energy. Observe the blocking artifact of the Haar wavelet



Compression with the Haar Wavelet

The next figure shows the same image analyzed with the wavelet sym6 and thresholding set so that 90.4% of the detail (at all levels) set to zero and 99.76% of the image energy is retained.



Compression with sym6

Observe the denoizing effect of compression.

Assignment 4(optional)

1. Make a record of you own voice (reading a passage in a book for example).
2. Import the voice signal in Matlab.
3. Have Matlab play the signal to check that it is correctly imported.
4. Apply the discrete wavelet transform to the voice signal.
5. Compress the voice signal using wavelet shrinkage with soft thresholding and hard thresholding.
6. Compare between the results of using various wavelets and biorthogonal wavelets.
7. Report on the best wavelet that seems to give the highest level of compression before the reconstructed signal becomes unintelligible.