The Haar Transform*

Nick Kingsbury

Abstract

This module introduces the Haar transform.

Probably the simplest useful energy compression process is the Haar transform. In 1-dimension, this transforms a 2-element vector $\begin{pmatrix} x(1) & x(2) \end{pmatrix}^T$ into $\begin{pmatrix} y(1) & y(2) \end{pmatrix}^T$ using:

$$\begin{pmatrix} y(1) \\ y(2) \end{pmatrix} = T \begin{pmatrix} x(1) \\ x(2) \end{pmatrix} \tag{1}$$

where $T = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. Thus $y(1)$ and $y(2)$ are simply the sum and difference of $x(1)$ and $x(2)$, scaled by $\frac{1}{\sqrt{2}}$ to preserve energy.

Note that T is an orthonormal matrix because its rows are orthogonal to each other (their dot products are zero) and they are normalised to unit magnitude. Therefore $T^{-1} = T^T$. (In this case T is symmetric so $T^T = T$.) Hence we may recover x from y using:

$$\begin{pmatrix} x(1) \\ x(2) \end{pmatrix} = T^T \begin{pmatrix} y(1) \\ y(2) \end{pmatrix} \tag{2}$$

In 2-dimensions x and y become 2×2 matrices. We may transform first the columns of x, by premultiplying by T, and then the rows of the result by postmultiplying by T^T. Hence:

$$y = TxT^T \tag{3}$$

and to invert:

$$x = T^T yT \tag{4}$$

To show more clearly what is happening:

If

$$x = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

*Version 2.6: Jun 8, 2005 10:19 am +0000
†http://creativecommons.org/licenses/by/1.0

then

$$y = \frac{1}{2}\begin{pmatrix} a+b+c+d & a-b+c-d \\ a+b-c-d & a-b-c+d \end{pmatrix}$$

These operations correspond to the following filtering processes:

- **Top left:** $a+b+c+d$ = 4-point average or 2-D lowpass (Lo-Lo) filter.
- **Top right:** $a-b+c-d$ = Average horizontal gradient or horizontal highpass and vertical lowpass (Hi-Lo) filter.
- **Lower left:** $a+b-c-d$ = Average vertical gradient or horizontal lowpass and vertical highpass (Lo-Hi) filter.
- **Lower right:** $a-b-c+d$ = Diagonal curvature or 2-D highpass (Hi-Hi) filter.

To apply this transform to a complete image, we group the pels into 2×2 blocks and apply (3) to each block. The result (after reordering) is shown in Figure 1(b). To view the result sensibly, we have grouped all the top left components of the 2×2 blocks in y together to form the top left subimage in Figure 1(b), and done the same for the components in the other 3 positions to form the corresponding other 3 subimages.

(a) (b)

Figure 1: Original Figure 1(a) and the Level 1 Haar transform Figure 1(b) of the 'Lenna' image.

It is clear from Figure 1(b) that most of the energy is contained in the top left (Lo-Lo) subimage and the least energy is in the lower right (Hi-Hi) subimage. Note how the top right (Hi-Lo) subimage contains the near-vertical edges and the lower left (Lo-Hi) subimage contains the near-horizontal edges.

The energies of the subimages and their percentages of the total are:

Lo-Lo	**Hi-Lo**	**Lo-Hi**	**Hi-Hi**
201.73×10^6	4.56×10^6	1.89×10^6	0.82×10^6
96.5%	2.2%	0.9%	0.4%

Table 1

Total energy in Figure 1(a) and Figure 1(b) = 208.99×10^6.

We see that a significant compression of energy into the Lo-Lo subimage has been achieved. However the energy measurements do not tell us directly how much data compression this gives.

A much more useful measure than energy is the **entropy** of the subimages after a given amount of quantisation. This gives the minimum number of bits per pel needed to represent the quantised data for each subimage, to a given accuracy, assuming that we use an ideal entropy code. By comparing the total entropy of the 4 subimages with that of the original image, we can estimate the compression that one level of the Haar transform can provide.