

# MCIT 515

## Fundamentals of Linear Algebra and Optimization Jean Gallier and Jocelyn Quaintance

### Project 2B: Compression of 2D images using Haar wavelets

The purpose of this project is to investigate properties and applications of Haar wavelets. In particular, methods for compressing images are investigated.

(Part 1) (**40 points**) Matrices can be compressed using the Haar transform as follows. Given a  $2^m \times 2^n$  matrix  $A$ , we can first convert the *rows* of  $A$  to their Haar coefficients using the Haar transform  $W_n^{-1}$  (where  $W_m$  is the  $2^m \times 2^m$  matrix whose columns are the vectors in the Haar basis of dimension  $2^m$ ), obtaining a matrix  $B$ , and then convert the *columns* of  $B$  to their Haar coefficients, using the matrix  $W_m^{-1}$ . Because columns and rows are exchanged in the first step,

$$B = A(W_n^{-1})^\top,$$

and in the second step  $C = W_m^{-1}B$ , thus, we have

$$C = W_m^{-1}A(W_n^{-1})^\top.$$

In the other direction, given a  $2^m \times 2^n$  matrix  $C$  of Haar coefficients, we reconstruct the matrix  $A$  (the image) by first applying  $W_m$  to the columns of  $C$ , obtaining  $B$ , and then  $W_n^\top$  to the rows of  $B$ . Therefore

$$A = W_m C W_n^\top.$$

Of course, we don't actually have to invert  $W_m$  and  $W_n$  and perform matrix multiplications. We just have to use our algorithms using averaging and differencing.

Write two `Matlab` functions `haar2D` and `haar_inv2D` implementing the method for computing the Haar transform of a matrix and the reconstruction of an image from its matrix of Haar coefficients, as described above. These will be tested using the auto-grader using the output script.

In the report, apply the function `haar_inv2D` to the matrix

$$T = \begin{pmatrix} 1212 & -306 & -146 & -54 & -24 & -68 & -40 & 4 \\ 30 & 36 & -90 & -2 & 8 & -20 & 8 & -4 \\ -50 & -10 & -20 & -24 & 0 & 72 & -16 & -16 \\ 82 & 38 & -24 & 68 & 48 & -64 & 32 & 8 \\ 8 & 8 & -32 & 16 & -48 & -48 & -16 & 16 \\ 20 & 20 & -56 & -16 & -16 & 32 & -16 & -16 \\ -8 & 8 & -48 & 0 & -16 & -16 & -16 & -16 \\ 44 & 36 & 0 & 8 & 80 & -16 & -16 & 0 \end{pmatrix}.$$

Compare your result with the matrix  $P_{bad}$  of Example 4.1 of the paper by Greg Ames (see the web page for CIS515) shown below:

$$P_{bad} = \begin{pmatrix} 576 & 704 & 1152 & 1280 & 1344 & 1472 & 1536 & 1536 \\ 704 & 640 & 1156 & 1088 & 1344 & 1408 & 1536 & 1600 \\ 768 & 832 & 1216 & 1472 & 1472 & 1536 & 1600 & 1600 \\ 832 & 832 & 960 & 1344 & 1536 & 1536 & 1600 & 1536 \\ 832 & 832 & 960 & 1216 & 1536 & 1600 & 1536 & 1536 \\ 960 & 896 & 896 & 1088 & 1600 & 1600 & 1600 & 1536 \\ 768 & 768 & 832 & 832 & 1280 & 1472 & 1600 & 1600 \\ 448 & 768 & 704 & 640 & 1280 & 1408 & 1600 & 1600 \end{pmatrix}$$

The matrix in Ames's paper seems to have a typo! What is it?

(Part 2) (**20 points**) You can load and display various images in **Matlab** using the following lines of code:

```
clear X map
load('durer','X')
Xdurer = X(1:512,:);
Xdurer(:,510:512) = 50;
figure
colormap(gray)
imagesc(Xdurer)
```

The above loads the file **durer**. There are a few other images such as **detail**, **flujet**, **earth**, **mandrill**, **spine**, and **clown**. You may have to resize these images to have dimensions that are powers of 2. To display an image, use **imagesc**.

Convert **Xdurer** to its Haar transform and decode it. Compare the original and the reconstructed image. Do the same with the other images **detail**, **flujet**, **earth**, **mandrill**, **spine**, and **clown**. Show the transformed version of **Xdurer** (and the other images if you so choose). Examples of Haar images from a different image example are in figures 1 and 2 (your color scheme may differ from the examples depending which colormap you choose to use).

What happens if you set small (in absolute value) coefficients of the transformed image to zero? Test at a few thresholds.

(Part 3) (**40 points**) Recall that  $W_n^\top$  is not the inverse of  $W_n$ , but rather the matrix

$$W_n^{-1} = D_n W_n^\top$$

$$\text{with } D_n = \text{diag}\left(2^{-n}, \underbrace{2^{-n}}_{2^0}, \underbrace{2^{-(n-1)}, 2^{-(n-1)}}_{2^1}, \underbrace{2^{-(n-2)}, \dots, 2^{-(n-2)}}_{2^2}, \dots, \underbrace{2^{-1}, \dots, 2^{-1}}_{2^{n-1}}\right).$$

The orthogonal matrix

$$H_n = W_n D_n^{\frac{1}{2}}$$

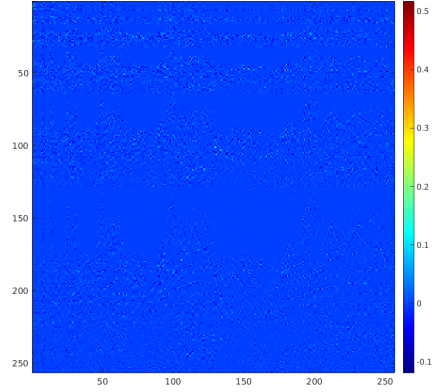


Figure 1: Example image and its haar transform (colored using colormap jet)

whose columns are the normalized Haar basis vectors, with

$$D_n^{\frac{1}{2}} = \text{diag}\left(2^{-\frac{n}{2}}, \underbrace{2^{-\frac{n}{2}}}_{2^0}, \underbrace{2^{-\frac{n-1}{2}}, 2^{-\frac{n-1}{2}}}_{2^1}, \underbrace{2^{-\frac{n-2}{2}}, \dots, 2^{-\frac{n-2}{2}}}_{2^2}, \dots, \underbrace{2^{-\frac{1}{2}}, \dots, 2^{-\frac{1}{2}}}_{2^{n-1}}\right)$$

is called the *normalized Haar transform matrix*. Given a vector (signal)  $u$ , we call  $c = H_n^\top u$  the *normalized Haar coefficients* of  $u$ .

Because  $H_n$  is orthogonal,  $H_n^{-1} = H_n^\top$ .

Then a moment of reflection shows that we have to slightly modify the algorithms to compute  $H_n^\top u$  and  $H_n c$  as follows: When computing the sequence of  $u^j$ s, use

$$\begin{aligned} u^{j+1}(2i-1) &= (u^j(i) + u^j(2^j+i))/\sqrt{2} \\ u^{j+1}(2i) &= (u^j(i) - u^j(2^j+i))/\sqrt{2}, \end{aligned}$$

and when computing the sequence of  $c^j$ s, use

$$\begin{aligned} c^j(i) &= (c^{j+1}(2i-1) + c^{j+1}(2i))/\sqrt{2} \\ c^j(2^j+i) &= (c^{j+1}(2i-1) - c^{j+1}(2i))/\sqrt{2}. \end{aligned}$$

Note that things are now more symmetric, at the expense of a division by  $\sqrt{2}$ . However, for long vectors, it turns out that these algorithms are numerically more stable.

Write two **Matlab** functions `haar2D_n` and `haar_inv2D_n` implementing the method for computing the normalized Haar transform of a matrix and the reconstruction of an image from its matrix of normalized Haar coefficients, as described above. These will be tested in the auto-grader as usual.

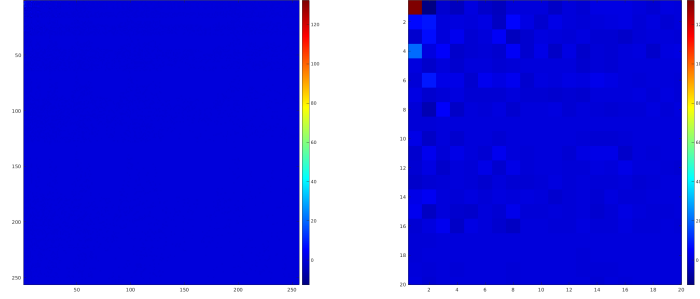


Figure 2: Example image from figure 1 and a representation of its normalized haar transform, with a zoomed in the upper right corner (colored using colormap jet)

Consider the image given by the following matrix:

$$A = \begin{pmatrix} 100 & 103 & 99 & 97 & 93 & 94 & 78 & 73 \\ 102 & 97 & 100 & 111 & 113 & 104 & 96 & 82 \\ 99 & 109 & 104 & 95 & 93 & 92 & 88 & 76 \\ 114 & 104 & 99 & 102 & 93 & 82 & 74 & 74 \\ 96 & 91 & 91 & 87 & 79 & 78 & 77 & 76 \\ 90 & 88 & 83 & 78 & 77 & 74 & 76 & 76 \\ 92 & 81 & 73 & 72 & 69 & 65 & 66 & 62 \\ 75 & 70 & 69 & 65 & 60 & 55 & 61 & 65 \end{pmatrix}$$

In the report, use `haar2D_n` to compute the normalized matrix  $C$  of Haar coefficients of  $A$ .

It is claimed in Ames's paper (Section 7) that the reconstructed matrix

$$A_2 = \begin{pmatrix} 100 & 100 & 95 & 95 & 92 & 92 & 76 & 76 \\ 103 & 103 & 98 & 98 & 106 & 106 & 90 & 90 \\ 99 & 109 & 99 & 99 & 96 & 96 & 81 & 81 \\ 114 & 104 & 104 & 104 & 91 & 91 & 76 & 76 \\ 91 & 91 & 86 & 86 & 76 & 76 & 76 & 76 \\ 91 & 91 & 86 & 86 & 76 & 76 & 76 & 76 \\ 82 & 82 & 76 & 76 & 66 & 66 & 66 & 66 \\ 74 & 74 & 69 & 69 & 58 & 58 & 59 & 59 \end{pmatrix}$$

is obtained from the normalized matrix

$$C_1 = \begin{pmatrix} 255 & 52 & 15 & 21 & 0 & 0 & 0 & 0 \\ 78 & 0 & 0 & 22 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 38 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 11 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

but this not quite correct. First, the coefficient 255 should be 682, and other nonzero entries are missing. In your report, find the matrix  $C_2$ , a compressed version of  $C$  that gives back  $A_2$ . The matrix  $C_2$  is obtained by applying the following steps:

1. Apply `haar2D_n` to  $A$ , obtaining  $C_0$ .
2. Apply the command `round` to  $C_0$  to obtain a matrix  $C_1$  with integer entries.
3. Set to zero all entries of absolute value strictly less than 10 in  $C_1$  to obtain  $C_2$ .

Show the output of each of these steps in the report.