Digital Image Processing

Image Transforms

The Walsh/Hadamard Transform

DR TANIA STATHAKI

READER (ASSOCIATE PROFFESOR) IN SIGNAL PROCESSING
IMPERIAL COLLEGE LONDON

Imperial College London

Introduction to Discrete Walsh Transform

- **Welcome back to the Digital Image Processing lecture!**
- In this lecture we will learn about the Discrete Walsh Transform (DWT) and the Discrete Hadamard Transform (DHT) in images.
- These two transforms are very similar. We often use the term Walsh-Hadamard transform.
- These transforms are different from the transforms we have met so far.
- Suppose we have the signal $f(x)$, $0 \le x \le N-1$ where $N = 2^n$. Note that the size of the signal is a power of 2.
- As with the case of Fourier transform, we transform our signal into a new domain where the independent variable is u.
- For both x and u we use the binary representation instead of the decimal one. We require n bits to represent both x and u.
- Therefore, we can write

  $(x)_{10} = (b_{n-1}(x) \ldots b_1(x)\, b_0(x))_2$ and $(u)_{10} = (b_{n-1}(u) \ldots b_1(u)\, b_0(u))_2$.

Introduction to Discrete Walsh Transform cont.

- Suppose that $f(x)$ has 8 samples.
- In that case $N = 8 = 2^n$ and therefore, $n = 3$.
- Note that the size of the signal is a power of 2.
- For $x = 6$ we use $(x)_2 = 110$. Note that the subscript 2 in $(x)_2$ indicates binary representation for x
- Therefore, in this case $b_2(x) = 1$, $b_1(x) = 1$ $b_0(x) = 0$.

Imperial College London

Definition: One-dimensional Discrete Walsh Transform (1D DWT)

- We will define now the 1D Discrete Walsh Transform as follows:

$$W(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)}$$

- The above is equivalent to:

$$W(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)(-1)^{\sum_{i=0}^{n-1} b_i(x)b_{n-1-i}(u)}$$

- The transform kernel values are obtained from:

$T(u,x) = \frac{1}{N}\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)} = \frac{1}{N}(-1)^{\sum_{i=0}^{n-1} b_i(x)b_{n-1-i}(u)}$

- Note that the transform kernel values are either 1 or -1.

Definition: One-dimensional Discrete Walsh Transform cont.

- We would like to write the Walsh transform in matrix form.
- We define the column vectors which contain the signal samples in both the original domain and Walsh transform domain as follows:

$$\underline{f} = \begin{bmatrix} f(0) \\ f(1) \\ \vdots \\ f(N-1) \end{bmatrix} \text{ and } \underline{W} = \begin{bmatrix} W(0) \\ W(1) \\ \vdots \\ W(N-1) \end{bmatrix}$$

- The Walsh transform can be written in matrix form as:

$$\underline{W} = T \cdot \underline{f}$$

- The elements $T(u, x)$ of matrix T of size $N \times N$ are defined in the previous slide. As already mentioned they are either 1 or -1.
- It can be shown immediately that the matrix T is a real, symmetric matrix with orthogonal columns and rows.
- We can easily show that $T^{-1} = N \cdot T^T = N \cdot T$.

Definition: One-dimensional Inverse Discrete Walsh Transform (IDWT)

- The Inverse Walsh transform is **almost identical** to the forward transform.

$$f(x) = \sum_{u=0}^{N-1} W(u) \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)}$$

- The above is equivalent to:

$$f(x) = \sum_{u=0}^{N-1} W(u) (-1)^{\sum_{i=0}^{n-1} b_i(x) b_{n-1-i}(u)}$$

- The matrix $I = T^{-1}$ formed by the inverse Walsh transform is identical to the one formed by the forward Walsh transform apart from a multiplicative factor N.
- In other words

$$I(u,x) = \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)} = N \frac{1}{N} \prod_{i=0}^{n-1} (-1)^{b_i(x) b_{n-1-i}(u)} = NT(u,x).$$

- This verifies the relation $I = T^{-1} = NT = N \cdot T^T$. Therefore, inverting T is an easy task.

Definition: Two-dimensional Discrete Walsh Transform (2D DWT) Forward and Inverse

- We define now the 2-D Walsh transform as a straightforward extension of the 1-D Walsh transform:

$$W(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y}^{N-1} f(x,y)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)}$$

- The above is equivalent to:

$$W(u,v) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y}^{N-1} f(x,y)(-1)^{\sum_{i=0}^{n-1}(b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v))}$$

- The inverse transform is **identical** to the forward as follows.

$$f(x,y) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y}^{N-1} W(u,v)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v)}$$

$$f(x,y) = \frac{1}{N}\sum_{x=0}^{N-1}\sum_{y}^{N-1} W(u,v)(-1)^{\sum_{i=0}^{n-1}(b_i(x)b_{n-1-i}(u)+b_i(y)b_{n-1-i}(v))}$$

Imperial College London

Properties of Two-Dimensional Discrete Walsh Transform

- The 2-D Walsh transform is separable and symmetric.
- Therefore it can be implemented as a sequence of two 1-D Walsh transforms, in a fashion similar to that of the 2-D DFT.
- Remember that the Fourier transform basis functions consist of complex sinusoids.
- The Walsh transform consists of basis functions whose values are only 1 and -1.
- They have the form of square waves.
- These functions can be implemented more efficiently in a digital environment than the exponential basis functions of the Fourier transform.
- For 1-D signals the forward and inverse Walsh kernels differ only in a constant multiplicative factor of N.
- This is because the array formed by the kernels is a symmetric matrix having **orthogonal** rows and columns, so its inverse array is almost the same as the array itself.
- In 2-D signals the forward and inverse Walsh kernels are identical.

Properties of Two-Dimensional Discrete Walsh Transform cont.

- The concept of frequency exists also in Walsh transform basis functions.
- We can think of frequency as the number of zero crossings or the number of transitions from positive to negative and vice versa in a basis vector and we call this number **sequency**.
- For the fast computation of the Walsh transform there exists an algorithm called **Fast Walsh Transform (FWT).** This is a straightforward modification of the FFT.

Definition: Two-Dimensional Discrete Hadamard Transform (2D-DHT) Forward and Inverse

- We define now the 2-D Hadamard transform. It is similar to the 2-D Walsh transform (look at subscripts of the circled bits below for the difference with Walsh).

$$H(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y}^{N-1}f(x,y)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_i(u)+b_i(y)b_i(v)}$$

- The above is equivalent to:

$$H(u,v)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y}^{N-1}f(x,y)(-1)^{\sum_{i=0}^{n-1}(b_i(x)b_i(u)+b_i(y)b_i(v))}$$

- The inverse transform is identical to the forward transform as follows.

$$f(x,y)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y}^{N-1}H(u,v)\prod_{i=0}^{n-1}(-1)^{b_i(x)b_i(u)+b_i(y)b_i(v)}$$

$$f(x,y)=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y}^{N-1}H(u,v)(-1)^{\sum_{i=0}^{n-1}(b_i(x)b_i(u)+b_i(y)b_i(v))}$$

Properties of Two-Dimensional Discrete Hadamard Transform

- Most of the comments made for Walsh transform are valid here.
- The Hadamard transform differs from the Walsh transform only in the order of basis functions. The order of basis functions of the Hadamard transform does not allow the fast computation of it by using a straightforward modification of the FFT.
- An important property of the Hadamard transform is that, letting H_N represent the Hadamard transform of order N the recursive relationship holds:

$$H_{2N} = \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

- Therefore, starting from a small Hadamard matrix we can compute a Hadamard matrix of any size.

Ordered Walsh and Hadamard Transforms

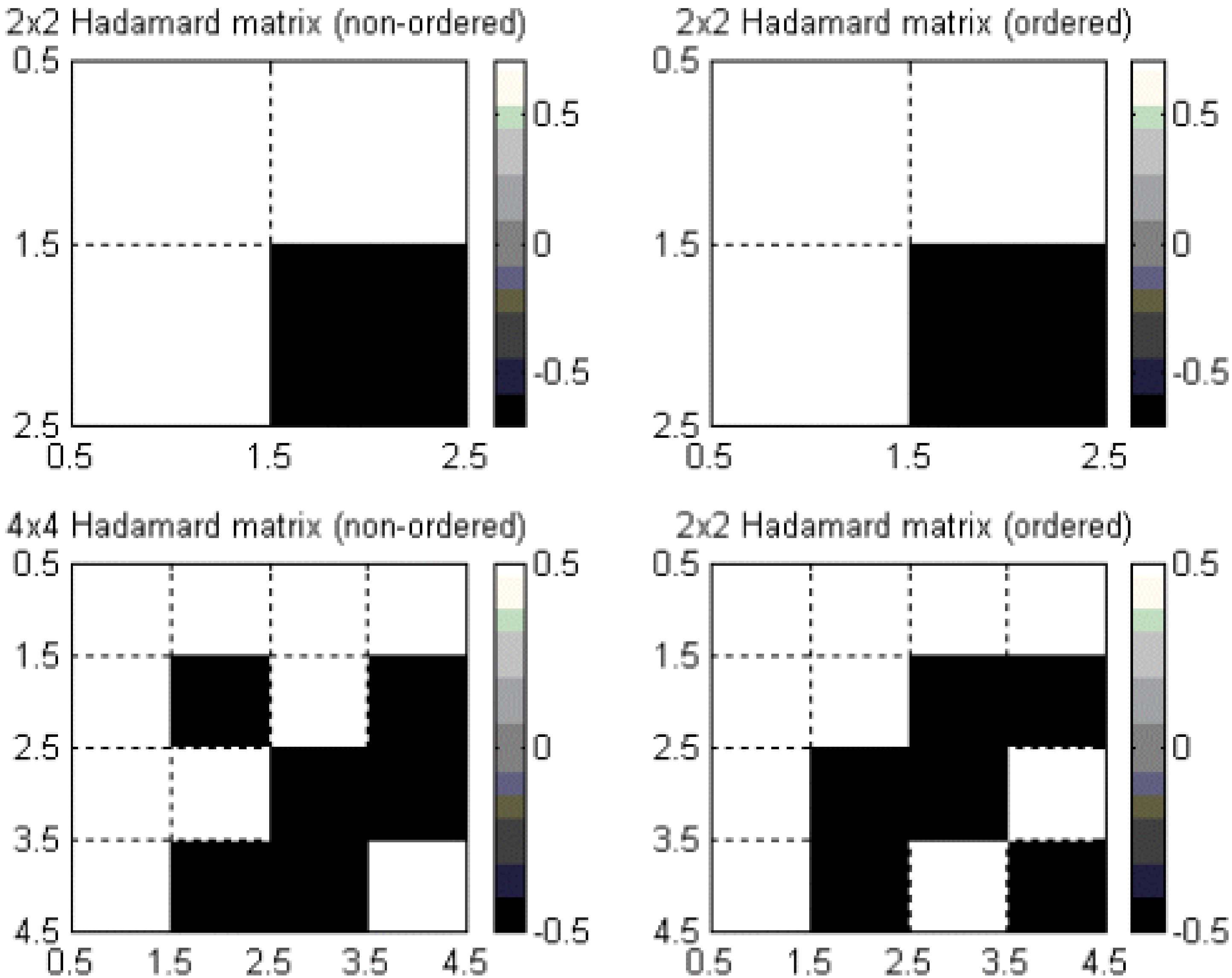
- Modified versions of the Walsh and Hadamard transforms can be formed by rearranging the rows of the transformation matrix so that the sequency increases as the index of the transform increases.

- These are called ordered transforms.

- The ordered Walsh/Hadamard transforms do exhibit the property of energy compaction whereas the original versions of the transforms do not. This observation is very prominent in the original Hadamard transform, where the basis functions are completely unordere, whereas in the original Walsh transform the basis functions are almost ordered.

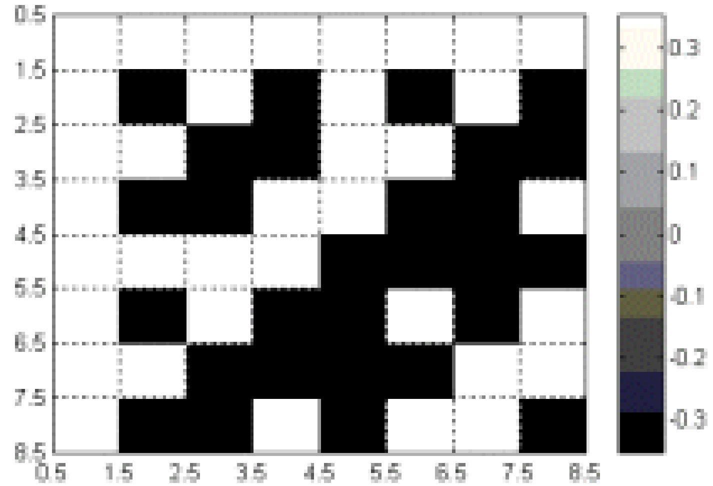
- Among all the transforms of this family, the Ordered Hadamard is the most popular due to its recursive matrix property.

Images of 1-D Hadamard matrices

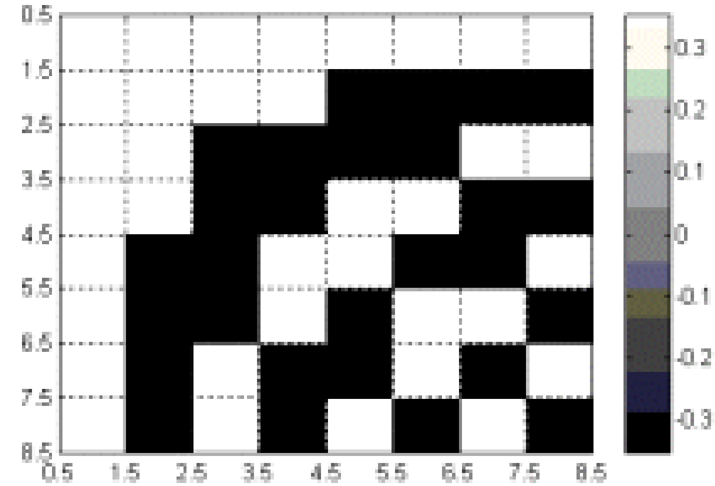


More images of 1-D Hadamard matrices

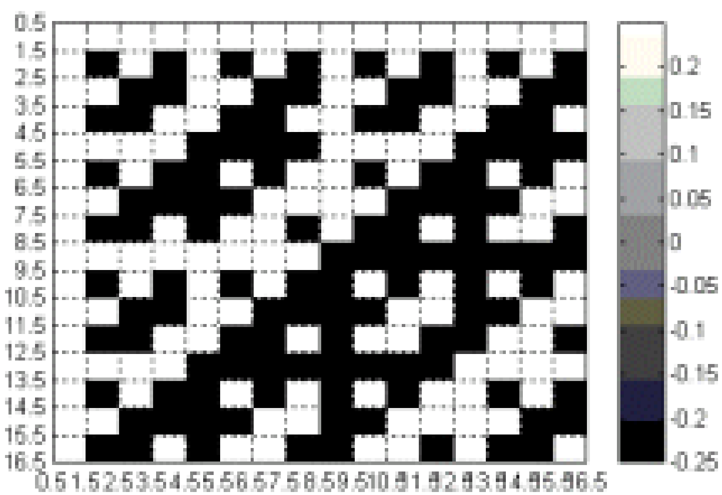
8x8 Hadamard matrix (non-ordered)



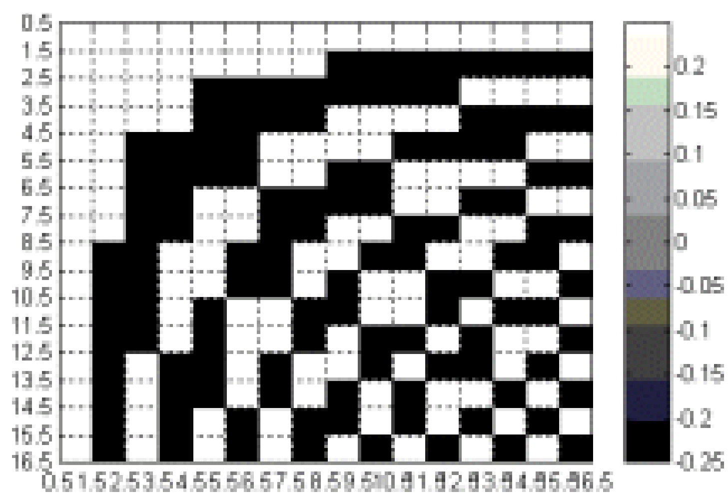
8x8 Hadamard matrix (ordered)



16x16 Hadamard matrix (non-ordered)

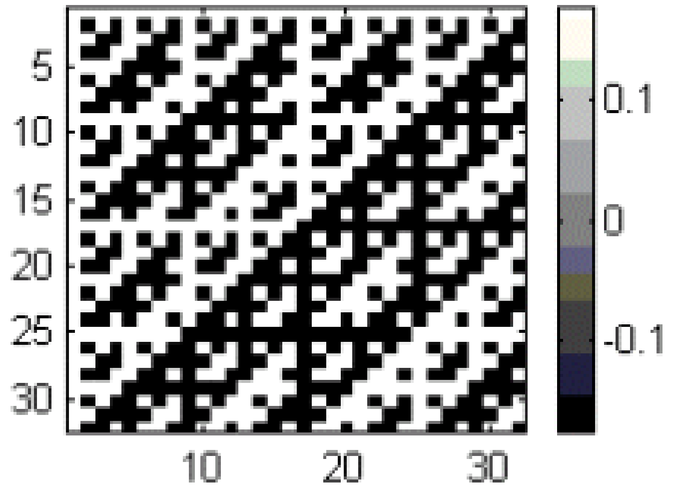


16x16 Hadamard matrix (ordered)

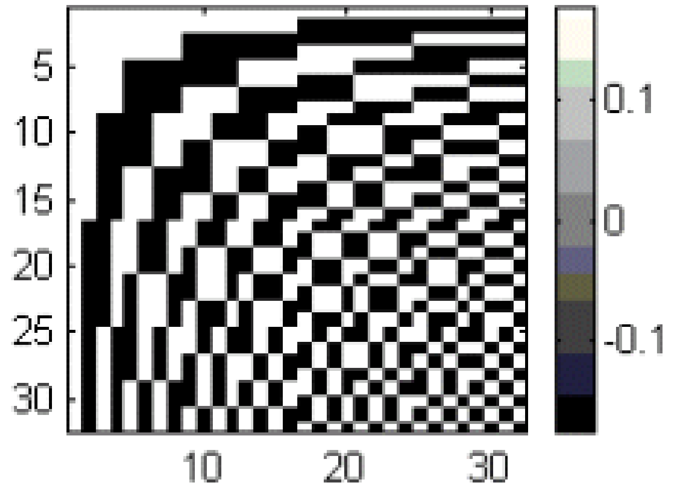


More images of 1-D Hadamard matrices

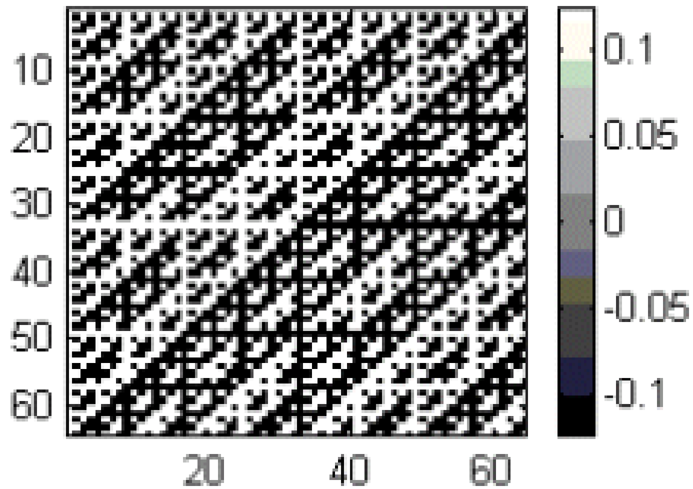
32x32 Hadamard matrix (non-ordered)



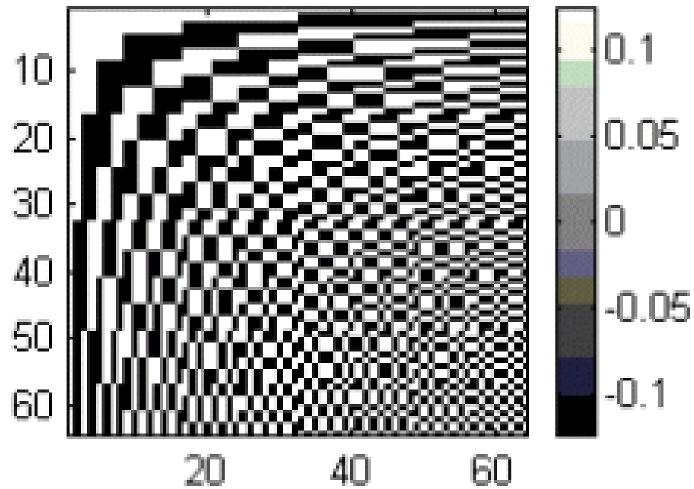
32x32 Hadamard matrix (ordered)



64x64 Hadamard matrix (non-ordered)



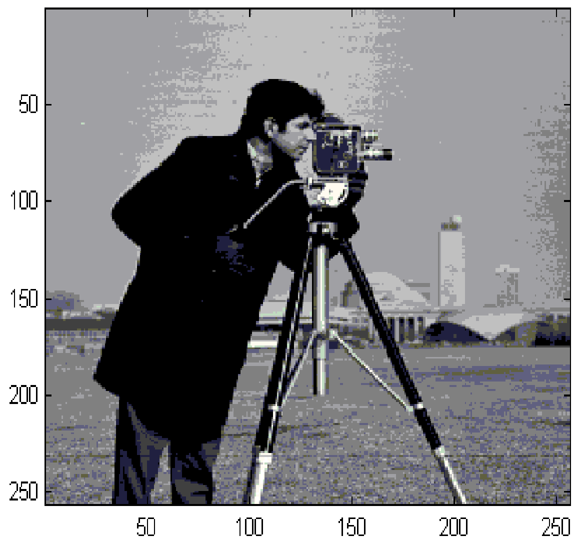
64x64 Hadamard matrix (ordered)



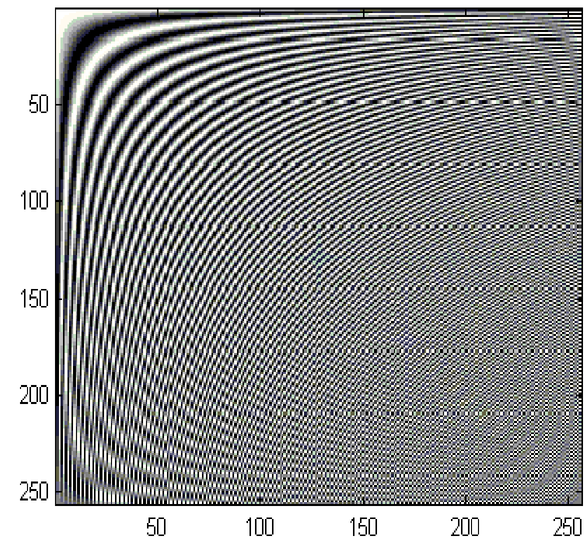
Experiment/Optional homework: Demonstrate and compare the energy compaction property of DCT and ordered Hadamard transform.

- Consider an image of size $N \times N$. The pixel $(0,0)$ is located on the top left corner.
- Calculate the DCT of the image.
- Take a small image patch of size $i \times i$ located on the top left part of the DCT transformed image. This patch contains the low frequencies of the original image.
- Calculate the fraction of the total image energy $e(i)$ that is contained on the patch of size $i \times i$.
- Repeat the above experiment for $i = 1, \dots, N$.
- You realise that the smallest possible patch is of size 1×1 (one DCT value only is kept; the $(0,0)$ value) and the largest possible patch is of size $N \times N$ (the entire DCT image is kept).
- Plot $e(i)$ as a function of i.
- Repeat the above experiment for the ordered Hadamard transform.
- The above experiment is depicted in the following slide.

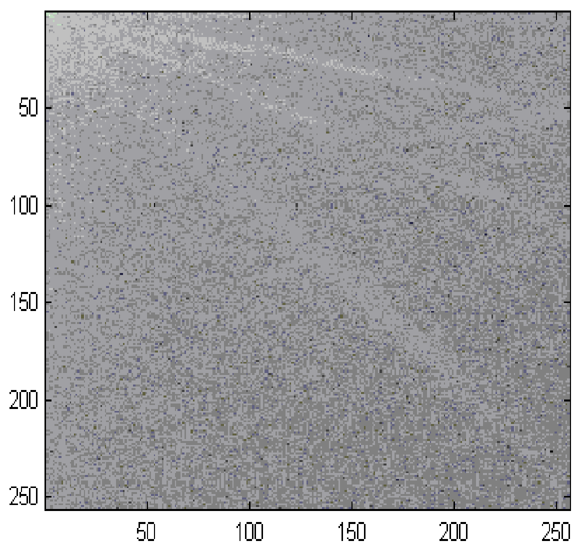
Superiority of DCT in terms of energy compaction in comparison to Hadamard



The 256x256 DCT matrix

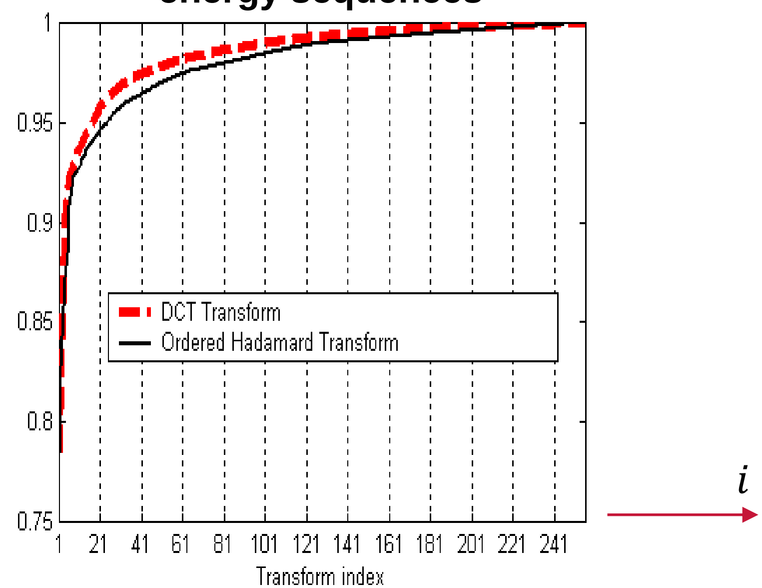


Display of a logarithmic function of the DCT of "cameraman"



$e(i)$

The cumulative energy sequences



Experiment: Demonstrate the energy compaction property of DCT and ordered Hadamard transform: Observations

- Obviously $e(i)$ is a monotonically increasing function. We can call it **cumulative energy** function.
- For a fixed i the percentage of energy contained in the DCT patch is higher than the percentage of energy contained in the Hadamard patch.

  Observe that the DCT curve is located slightly above the Hadamard curve.
- Or, in order to keep a fixed percentage of the total energy of the image, you require a smaller patch when you use DCT compared to the one you require if you use Hadamard.
- In the above experiments you observe that as i increases from 1 to slightly higher values the fraction of total energy $e(i)$ becomes almost 1 quite quickly. In other words, the preserved patch contains almost 100% of the energy of the entire DCT image.
- Don't forget the energy preservation property; both space and transformed image have identical energies.

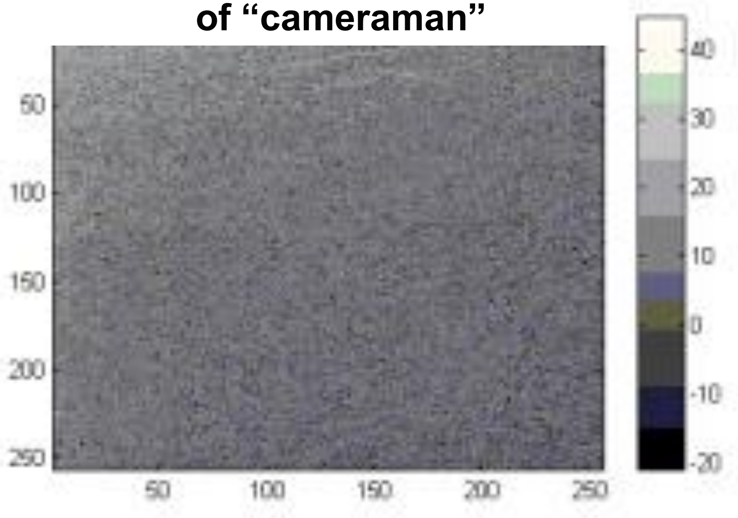
Experiment: Demonstrate the energy compaction property of DCT and ordered Hadamard transform: Observations cont.

- The observations in the previous two slides rely on the fact that both DCT and Ordered Hadamard transform exhibit the property of energy compaction.
- When we repeat the experiment for the non-ordered Hadamard transform we observe that the cumulative energy function does not increase quickly towards its maximum value. On the contrary, it looks more or less like the straight line type of function with moderate slope. To verify this observation, look at the next slide, bottom left figure.

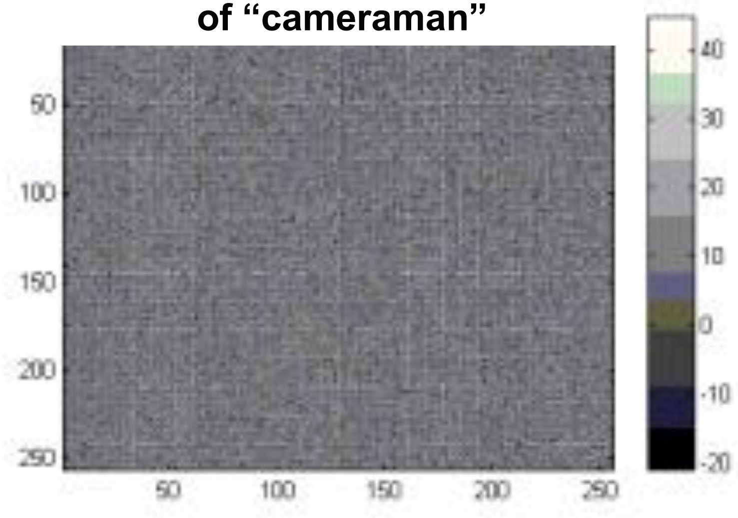
Imperial College London

Non-ordered and ordered Hadamard

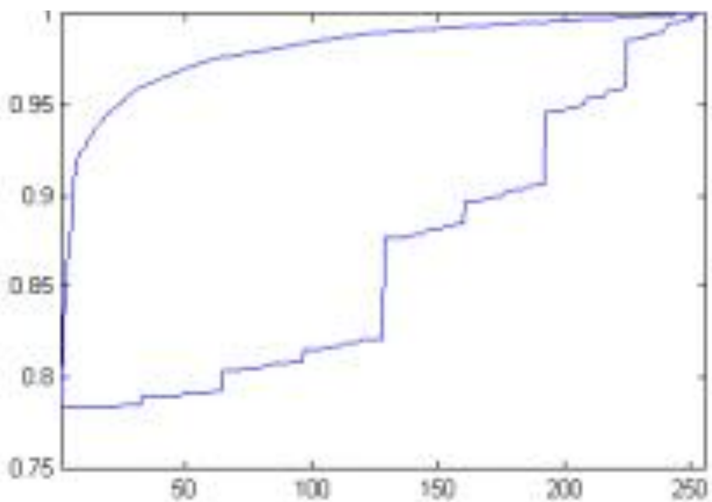
The ordered Hadamard Transform of “cameraman”



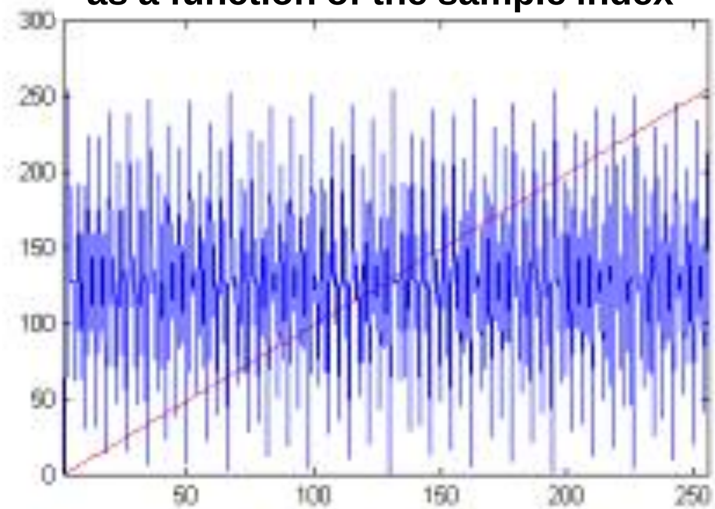
The non-ordered Hadamard Transform of “cameraman”



The cumulative transform energy sequences



The sequency of transform as a function of the sample index



Question: In the bottom figures which of the two transforms is related to each curve?

2-D Walsh Basis Functions

- 2-D Walsh basis functions for 4×4 images.
- x, y, u, v are between 0 and 3.
- For a fixed (u, v) we plot the 4×4 functions $T(u, x, v, y)$.