Homework 4 - Solutions

ELEG 404/604 - Digital Image and Audio Signal Processing
Department of Electrical and Computer Engineering
Spring 2016
Problem 1.a

1. Consider a 128 × 128 Hadamard transform.

   a. Using MatLab, display and print the transform matrix when ordered in sequency, normal Hadamard, and dyadic forms.
Hadamard transform

The Hadamard matrices of dimension $2^k$ for $k \in \mathbb{N}$ are given by the recursive formula:

The lowest order of Hadamard matrix is 2

$$H(2^1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H(2^2) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

And in general

$$H(2^k) = \begin{bmatrix} H(2^{k-1}) & H(2^{k-1}) \\ H(2^{k-1}) & -H(2^{k-1}) \end{bmatrix} = H(2) \otimes H(2^{k-1})$$
Normal Order

\[ k = 1 \rightarrow H(2) \]

\[ H(2^k) = H(2) \otimes H(2^{k-1}) \]
Kronnecker Product

The **Kronecker product** of two matrices \( A = [a_{ij}]_{m \times n} \) and \( B = [b_{ij}]_{k \times l} \) is defined as

\[
A \otimes B \triangleq \begin{bmatrix}
    a_{11}B & \cdots & a_{1n}B \\
    \vdots & \ddots & \vdots \\
    a_{m1}B & \cdots & a_{mn}B
\end{bmatrix}_{mk \times nl}
\]

```matlab
function [ C ] = kronprod( A,B )
%Kronecker Product
[x1,y1]=size(A);
[x2,y2]=size(B);
C=zeros(x1*x2,y1*y2);
k=1;
l=1;

for i=1:1:x1
    for j=1:1:y1
        C(k:k+x2-1,l:l+y2-1)=A(i,j)*B;
        l=l+y2;
    end
    k=k+x2;
    l=1;
end
```

4/11/2016
MRI - Spring 2016
Normal ordering

Hadamard Transform (Normal)
Sequence ordering

- The sequency ordering can be derived from the ordering of the Hadamard matrix by first applying the **Gray code permutation** and the **bit-reversal permutation**.

### Bit-reversal permutation

<table>
<thead>
<tr>
<th>Original</th>
<th>Reversed</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>01</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>01</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>
Gray code permutation

Binary numeral system where two successive values differ in only one bit (binary digit).

<table>
<thead>
<tr>
<th>2-bit</th>
<th>3-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>000</td>
</tr>
<tr>
<td>01</td>
<td>001</td>
</tr>
<tr>
<td>11</td>
<td>011</td>
</tr>
<tr>
<td>10</td>
<td>010</td>
</tr>
<tr>
<td>110</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Sequence ordering

% a.1 Ordered in seqeuncy

figure('units','normalized','position',[0 0 1 1])
Hseq = walsh(128);
imagesc(Hseq);
colormap('gray');
title('Hadamard Transform. Ordered in Sequence');

% Generate the Hadamard matrix
H = hadamard(N);

% generate Gray code of size N.
graycode = [0;1];
while size(graycode,1) < N
    graycode = kron([0;1], ones(size(graycode,1),1)) +
              [graycode; flipud(graycode)];
end

% Generate indices from bit-reversed Gray code.
seqord = bin2dec(fliplr(char(graycode+'0')))+1;

% Reorder H.
H = H(seqord,:);
Sequence ordering

Hadamard Transform. Ordered in Sequence
Dyadic ordering

Also known as Paley order. It is related to the sequence ordering by Gray code reordering of the rows, and to the **Normal Ordering by bit reversal of its rows**.

```matlab
function D = mtxdya(N)

% Generate the Hadamard matrix
H = hadamard(N);

% Get binary indices and flip them to reorder the
% Hadamard matrix
x=0:N-1;
x=x';
x=dec2bin(x);
x=fliplr(x);
x=bin2dec(x)+1;

% Reorder normal Hadamard matrix
D=H(x,:);
end
```

- Obtain binary indices
- Flip the binary values (reverse Order)
- Obtain decimal indices, and add 1 for Matlab indexing.
Dyadic ordering
Problem 1.b

1. Consider a $128 \times 128$ Hadamard transform.

   b. Consider the $128 \times 128$ image available in the attached file. Consider the spatial image domain $[x, y]$. Obtain the Hadamard transform of the image using the three different ordered Hadamard matrices.
Padding

% Load the image
load ('img.mat');

% Obtaining the transform with the Hadamard transform ordered by sequence
% Y= H X H , Y is the transform, X is the original image, before computing
% the transform, we have to zero pad both the image and the transformation
% matrix to avoid errors in the reconstruction:

img1=zeros(256,256);
img1(1:128,1:128)=img;

Hseqzp=zeros(256,256);
Hnormzp=zeros(256,256);
Hdyazp=zeros(256,256);

Hseqzp(1:128,1:128)=Hseq;
Hnormzp(1:128,1:128)=Hnorm;
Hdyazp(1:128,1:128)=Hdy;
Transform

```matlab
% Obtaining the transform with the Hadamard transform (sequence)
% Y = H * X * H, Y is the transform, X is the original image:

Y1=(1/sqrt(2^7)* Hseqzp) * img1 *(1/sqrt(2^7)* Hseqzp);
Ylim=Y1(1:128,1:128);
Ylim=Ylim-(min(min(Ylim)));
Ylim=round(255*Ylim/(max(max(Ylim))));

figure
imshow(Ylim,[]);
title('Hadamard Transform. Ordered in Sequence');
```
Hadamard Transform. (Normal)
Hadamard Transform. Ordered in Sequence
Hadamard Transform. (Dyadic Form)
Problem 1.c

1. Consider a $128 \times 128$ Hadamard transform.

c. Compute the percentage of the energy packed in each of the quadrants shown in the figure.
Case: Normal

Percentage of the first quadrant
\[ \text{Perc1\_norm} = 83.1297\% \]

Percentage of the second quadrant
\[ \text{Perc2\_norm} = 7.4685\% \]

Percentage of the third quadrant
\[ \text{Perc3\_norm} = 7.8350\% \]

Percentage of the fourth quadrant
\[ \text{Perc4\_norm} = 1.5668\% \]
Case: Ordered Sequence
Percentage of the first quadrant
Perc1_sq = 99.1271

Percentage of the second quadrant
Perc2_sq = 0.5061

Percentage of the third quadrant
Perc3_sq = 0.3211

Percentage of the fourth quadrant
Perc4_sq = 0.0457

Case: Dyadic
Percentage of the first quadrant
Perc1_dya = 99.1271

Percentage of the second quadrant
Perc2_dya = 0.5061

Percentage of the third quadrant
Perc3_dya = 0.3211

Percentage of the fourth quadrant
Perc4_dya = 0.0457
Problem 1.d

1. Consider a 128 × 128 Hadamard transform.

d. Reconstruct and print the image when the coefficients in the upper left triangle of the transform plane are used (low-pass sequency).
triu(X) is the upper triangular part of X. In this case X is a 128x128 matrix of ones.

Pointwise product of the upper triangle matrix with the Hadamard Transforms.
triu(X) is the upper triangular part of X. In this case X is a 128x128 matrix of ones.

Pointwise product of the upper triangle matrix with the Hadamard Transforms.

Inverse Hadamard Transform
Reconstruction Hadamard Transform upper triangle. (Normal)
Reconstruction Hadamard Transform upper triangle. (Ordered Sequence)
Reconstruction Hadamard Transform upper triangle. (Dyadic Form)
Problem 1.e

1. Consider a 128 × 128 Hadamard transform.

   e. Reconstruct and print the image when the coefficients in the lower right triangle of the transform plane are used (high pass sequency).
triu(X) is the upper triangular part of X. In this case X is a 128x128 matrix of ones.

Pointwise product of the lower triangle matrix with the Hadamard Transforms.

Inverse Hadamard Transform
Reconstruction Hadamard Transform lower triangle. (Normal)
Reconstruction Hadamard Transform lower triangle. (Ordered Sequence)
Reconstruction Hadamard Transform lower triangle. (Dyadic Form)
1. Consider a $128 \times 128$ Hadamard transform.

f. Repeat (e) when only the top left quadrant (low pass sequence) coefficients are used.
% First we create a matrix with ones in the positions corresponding to the 
% first quadrant, zeros in the rest
Quad1 = zeros(128,128);
Quad1(1:64,1:64) = ones(64,64);

% Now we obtain the three new matrices of the transform plane:
Quad1HSeqO= zeros(256,256);
Quad1HNorm= zeros(256,256);
Quad1Hdyad= zeros(256,256);
Quad1HSeqO(1:128,1:128) = Quad1.*(Y1esc);
Quad1HNorm(1:128,1:128) = Quad1.*(Y2esc);
Quad1Hdyad(1:128,1:128) = Quad1.*(Y3esc);

% Reconstruction of the Images
RecHseqQuad11=(1/sqrt(2^7)*Hseqzp)*Quad1HSeqO*(1/sqrt(2^7)*Hseqzp);
RecHnormQuad11=(1/sqrt(2^7)*Hnormzp)*Quad1HNorm*(1/sqrt(2^7)*Hnormzp);
RecHdyadQuad11=(1/sqrt(2^7)*Hdyazp)*Quad1Hdyad*(1/sqrt(2^7)*Hdyazp);

Take the first 64x64 quadrant
Pointwise product of the upper left quadrant with the Hadamard Transforms.
Inverse Hadamard Transform
Reconstruction Hadamard Transform first quadrant. (Normal)
Reconstruction Hadamard Transform first quadrant. (Ordered Sequence)
Reconstruction Hadamard Transform first quadrant. (Dyadic Form)
Problem 2

2. Filter the given “characters.tif” image using Butterworth Lowpass and Highpass filters. Implement your own Butterworth functions and Display the results for orders $n = 2, 4$ and $Do = 10, 60, 460$. (i.e. DO NOT use imfilter or filter2)
Butterworth Low Pass Filter

Frequency response:

\[ H(u, v) = \frac{1}{1 + \left[\frac{D(u, v)}{D_0}\right]^{2n}} \]

- Order: \( n \), Cutoff frequency: \( D_0 \)
- Smooth transfer function
  - Minimizes ringing
  - Order controls transition bandwidth
% Load the image
charimg = imread('characters.tif');
charimg = im2double(charimg);

% First create the filters.
% Using the equation:
% D(u,v) = [(u-P/2)^2+(v-Q/2)^2]^(1/2), and
% the equation describing the filter:
% H(u,v) = 1/(1+(D(u,v)/Do)^2n)

n = 2; Do = 460

But2_460 = 1./(1+(Duv./460).^(2*2));

n = 4; Do = 10

But4_10 = 1./(1+(Duv./10).^(2*4));

n = 2; Do = 60

But4_60 = 1./(1+(Duv./60).^(2*4));

n = 2; Do = 460

But4_460 = 1./(1+(Duv./460).^(2*4));

% Obtain the padded image:
PadImg = zeros(2048,2048);
PadImg(1:size(charimg,1),1:size(charimg,2)) = charimg;

u = 0:1:2047;
v = 0:1:2047;

[U,V] = meshgrid(u,v);
Du = (U-2047/2).^2;
Dv = (V-2047/2).^2;
Duv = (Du+Dv).^(1/2);

But2_10 = 1./(1+(Duv./10).^(2*2));

n = 2; Do = 60

But2_60 = 1./(1+(Duv./60).^(2*2));
Obtain the padded image:
PadImg=zeros(2048,2048);
PadImg(1:size(charImg,1),1:size(charImg,2))=charImg;

x=0:1:2047;
y=0:1:2047;

[X,Y]=meshgrid(x,y);

A=(-1).^(X+Y);

Multiply by \((-1)^{(x+y)}\) to center the transform
PadImg=PadImg.*A;

Compute the 2D FFT of the computed image
DFTImg=fft2(PadImg);
Form the product with all the filters, obtain the inverse 2D FFT of the image, obtain the real part of it and multiply again for \((-1)^{(x+y)}\), and crop the matrix to get just the image of the original size:

\[\text{n}=2 \quad \text{Do}=10\]
\[
\text{Filter}_1=\text{But2}_{_10} \star \text{DFTImg};
\]
\[
\text{Recon}_1=\text{real}(\text{iift2}(\text{Filter}_1)) \star \text{A};
\]
\[
\text{Im}_1=\text{Recon}_1(1:\text{size}(\text{charimg},1),1:\text{size}(\text{charimg},2));
\]
\[
\text{Im}_1=\text{Im}_1-\text{min}(\text{min}(\text{Im}_1));
\]
\[
\text{Im}_1=\text{round}(255 \star \text{Im}_1/(\text{max}(\text{max}(\text{Im}_1))));
\]
Low Pass Butterworth Filter $n=2$ $D_o=10$
Low Pass Butterworth Filter $n=2$ $D_0=60$
Low Pass Butterworth Filter n=2 Do=460
Low Pass Butterworth Filter $n=4$ $D_0=10$
Low Pass Butterworth Filter n=4 Do=60
Low Pass Butterworth Filter $n=4$ $D_0=460$
High Pass Filters

Highpass filters. $D_0$ is the cutoff frequency and $n$ is the order of the Butterworth filter.

\[

table

<table>
<thead>
<tr>
<th>Ideal</th>
<th>Butterworth</th>
<th>Gaussian</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H(u, v) = \begin{cases} 1 &amp; \text{if } D(u, v) \leq D_0 \ 0 &amp; \text{if } D(u, v) &gt; D_0 \end{cases}$</td>
<td>$H(u, v) = \frac{1}{1 + \left[D_0/D(u, v)\right]^{2n}}$</td>
<td>$H(u, v) = 1 - e^{-D^2(u,v)/2D_0^2}$</td>
</tr>
</tbody>
</table>
\]

Graphs of the ideal, Butterworth, and Gaussian filters are shown below.
2. Filters (DFT) (HighPass)

\%n=2 \hspace{1em} \text{Do}=10

\text{Buth2}_10=1/(1+((10./Duv).^(2*2)));

\%n=2 \hspace{1em} \text{Do}=60

\text{Buth2}_60=1/(1+((60./Duv).^(2*2)));

\%n=2 \hspace{1em} \text{Do}=460

\text{Buth2}_460=1/(1+((460./Duv).^(2*2)));

\%n=4 \hspace{1em} \text{Do}=10

\text{Buth4}_10=1./((10./Duv).^(2*4));

\%n=2 \hspace{1em} \text{Do}=60

\text{Buth4}_60=1./((60./Duv).^(2*4));

\%n=2 \hspace{1em} \text{Do}=460

\text{Buth4}_460=1./((460./Duv).^(2*4));

\% Form the product with all the filters, obtain the inverse 2D FFT of the image, obtain the real part of it and multiply again for \((-1)^{(x+y)}\), and crop the matrix to get just the image of the original size:

\%n=2 \hspace{1em} \text{Do}=10

\text{Filter1h}=\text{Buth2}_10.*\text{DFTImg};
\text{Recon1h}=\text{real}(\text{ifft2}(\text{Filter1h})).*\text{A};
\text{Im1h}=\text{Recon1h}(1:\text{size}(\text{charimg},1),1:\text{size}(\text{charimg},2));
High Pass Butterworth Filter n=2 Do=10
High Pass Butterworth Filter n=2 Do=60
High Pass Butterworth Filter n=2 Do=460
High Pass Butterworth Filter n=4 Do=10
High Pass Butterworth Filter n=4 Do=60
High Pass Butterworth Filter $n=4$ $Do=460$