Problem Set 5

1. (DTFT and DFT and FFT)

(a) x.* exp(-j * omega * reshape([0:N-1], size(x)));

here x is $[x[0], x[1], \dots, x[N-1]]$

reshape([0:N-1], size(x)) gives out $[0, 1, \dots N-1]$

then -j * omega * reshape gives $[e^{-j\omega 0}, e^{-j\omega 1}, \dots e^{-j(N-1)\omega}]$

$\therefore$ x. exp(-j * omega * reshape([0:N-1], size(x)))

gives $[x[0]e^{-j\omega 0}, x[1]e^{-j\omega 1}, \dots x[N-1]e^{-j\omega(N-1)}]$

$\therefore$ and the sum of it gives a DTFT of $x[n]$

Actually, it can also be written as:

y = x * exp(-j * omega * reshape([0:N-1], size(x))');

because if x is a row vector

$$[x[0], x[1], \dots x[N-1]] * [e^{-j\omega 0}, e^{-j\omega 1}, \dots e^{-j\omega(N-1)}]^T$$

$$= \sum_{n=0}^{N-1} x[n] e^{-j\omega n}$$

(b) (c) (d) (e) see next pages with code and figure.

1(b).
filename: mydft.m

```
function y=mydft(x,k)
%make sure x is a row vector
N = length(x);
x = reshape(x,[1,N]);
y = x * exp(-1i*k*2*pi/N * (0:N-1)');
```

Use the above function to form a dft computation. **filename: dft.m**

```
function [m,phase,telapsed] = dft(x,mtype)
%function [m,phase,telapsed] = dft(x)
%if mtype is 0, magnitude type be decible, else be real number
%plot the magnitude , phase and time elapsed

N = length(x);
y = zeros(1,N);
tic;
tstart = tic;
for i = 1: N
    y(i) = mydft(x,i-1);
end
telapsed = toc(tstart);
if mtype ==0
    m = 10*log10(abs(y));
else
    m = abs(y);
end
phase = angle(y);
disp('my dft time elapsed');
disp(num2str(telapsed));
subplot(2,1,1)
plot((0:N-1)*2/N,m)
if mtype == 0
    ylabel('Magnitude (dB)')
else
    ylabel('Magnitude')
end
xlabel('Frequency ({\times} \pi rad/sample)')
subplot(2,1,2)
plot((0:N-1)*2/N,phase)
ylabel('Phase')
xlabel('Frequency ({\times} \pi rad/sample)')
```

1(c).
Use the function generated in (b) to do the comparison.(M = 10).
filename: solution_c.m

```
clear
%two sequence:
M = 10;
N = 2^M;
n= 0: N-1;
x1 = cos(pi * n/11);
x2 =double( n <= N/2-1);

%(c) compare dft and dtft
%the first sequence
%dft
figure
```

```
title('dft')
[m1,p1,t1]=dft(x1);
%dtft
figure
title('dtft')
freqz(x1,1,N);

%the second sequence
%dft
figure
title('dft')
[m2,p2,t2]=dft(x2);
%dtft
figure
title('dtft')
freqz(x2,1,N);

%(d)compare dft and fft
%the first sequence
%dft
figure
title('dft')
```

Sequence 1

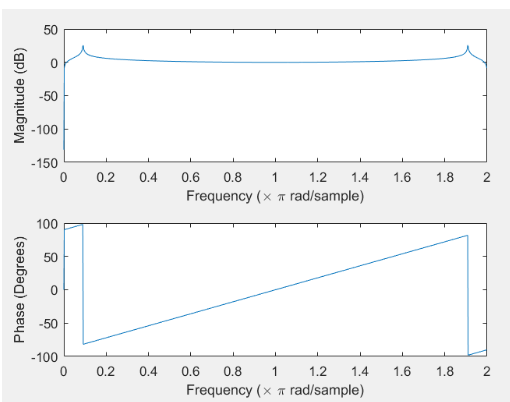


Figure 1: *dft*

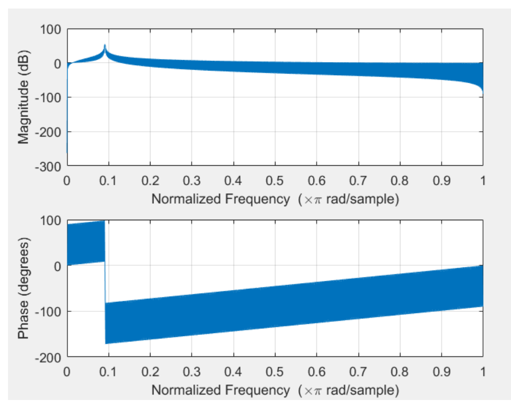


Figure 2: *dtft*

Sequence 2

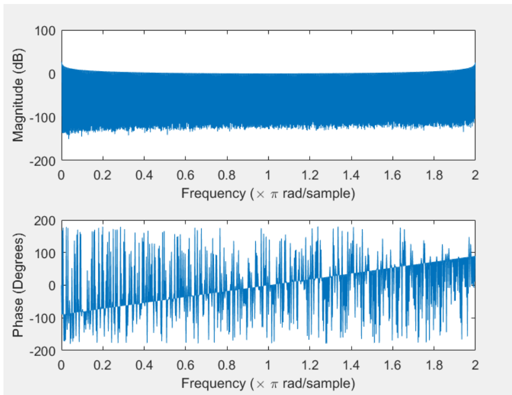


Figure 3: *dft*

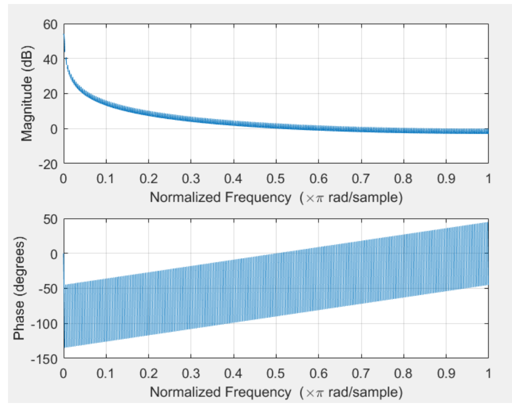


Figure 4: *dtft*

1(d) Some functions are created and running the following script to do the comparison.
filename: myfft.m

```
function [m,phase,telapsed]=myfft(x,mtype)
N = length(x);
tic;
tstart = tic;
y = fft(x);
telapsed = toc(tstart);
%plot information
if mtype ==0
    m = 10*log10(abs(y));
else
    m = abs(y);
end
phase = angle(y);
disp('my fft time elapsed');
disp(num2str(telapsed));
subplot(2,1,1)
plot((0:N-1)*2/N,m)
if mtype == 0
    ylabel('Magnitude (dB)')
else
    ylabel('Magnitude')
end
xlabel('Frequency ({\times} \pi rad/sample)')
subplot(2,1,2)
plot((0:N-1)*2/N,phase)
ylabel('Phase')
xlabel('Frequency ({\times} \pi rad/sample)')
```

filename:solution_d.m

```
%two sequence:
M = 17;
N = 2^M;
n= 0: N-1;
x1 = cos(pi * n/11);
x2 =double( n <= N/2-1);
```

```
%the first sequence
%dft choose dB
figure
[m1,p1,t1]=dft(x1,0);
%fft
figure
[m1_fft,p1_fft_t1_fft]=myfft(x1,0);

%the second sequence
%dft choose real number
figure
[m2,p2,t2]=dft(x2,1);
%fft
figure
[m2_fft,p2_fft_t2_fft]=myfft(x2,1);
```

Result:(M = 17)
my dft time elapsed
439.8895
my fft time elapsed
0.0098572
my dft time elapsed
437.2934
my fft time elapsed
0.002759
1(e)
Funvtion to generate one point. **filename: myConvn.m**

```
function y = myConvn(x1,x2,n)
length1 = length(x1);
length2 = length(x2);
sum = 0;
for m = 0 : length1-1
    if n-m>=0 && (n-m)<=length2-1
        sum = sum + x1(m+1)*x2(n-m+1);
    end
end
y=sum;
```

Function to generate convolution
filename:myConv.m

```
function y = myConv(x1,x2)
length1 = length(x1);
length2 = length(x2);
y = zeros(1,length1+length2);
tic;
tstart = tic;
for i = 0 : length1 + length2 - 1
    y(i+1) = myConvn(x1,x2,i);
end
telapsed = toc(tstart);
disp('time elapsed');
disp(num2str(telapsed));
```

Function for the convolution using fft
filename:fftConv.m

```
function x = fftConv(x1,x2)
N1 = length(x1);
N2 = length(x2);
x1 = [x1,zeros(1,N2-1)];
x2 = [x2,zeros(1,N1-1)];
tic;
tstart = tic;
X1 = fft(x1);
X2 = fft(x2);
X = X1 .* X2;
x = ifft(X);
telapsed = toc(tstart);
disp('fft convolve time elapsed');
disp(num2str(telapsed));
%fft convolution
```

Solution to this problem
filename: solution1e.m

```
clear
%two sequence:
M = 15;
N = 2^M;
n= 0: N-1;
x1 = cos(pi * n/11);
x2 =double( n <= N/2-1);

y1 = myConv(x1,x2);
y2 = fftConv(x1,x2);
```

Result:(M = 15)
regular convolve time elapsed
24.4747
fft convolve time elapsed
0.02541

2. You are given a finite-length signal $x[n]$, defined for $0 \le n \le N-1$, where $N = 3^M$ for some integer M. Derive a "radix-3 decimation-in-frequency" algorithm that computes the DFT of $x[n]$ in terms of the three length-$N/3$ DFTs of the signals.

Answer:

$$X[k] = \sum_{n=0}^{N/3-1} x[n] W_N^{kn} + \sum_{n=N/3}^{2N/3-1} x[n] W_N^{kn} + \sum_{n=2N/3}^{N-1} x[n] W_N^{kn}$$

$$= \sum_{n=0}^{N/3-1} x[n] W_N^{kn} + \sum_{n=0}^{N/3-1} x[n+N/3] W_N^{k(n+N/3)} + \sum_{n=0}^{N/3-1} x[n+2N/3] W_N^{k(n+2N/3)}$$

$$= \sum_{n=0}^{N/3-1} x[n] W_N^{kn} + W_N^{kN/3} \sum_{n=0}^{N/3-1} x[n+N/3] W_N^{kn} + W_N^{k2N/3} \sum_{n=0}^{N/3-1} x[n+2N/3] W_N^{kn}$$

$$= \sum_{n=0}^{N/3-1} \left(x[n] W_N^{kn} + x[n+N/3] W^{k/3} W_N^{kn} + x[n+2N/3] W^{2k/3} W_N^{kn} \right)$$

$$= \sum_{n=0}^{N/3-1} \left(x[n] + W^{k/3} x[n+N/3] + W^{2k/3} x[n+2N/3] \right) W_N^{kn}$$

split $X[k]$ into 3 part.

$$X[3k] = \sum_{n=0}^{N/3-1} \left(x[n] + W^{k} x[n+N/3] + W^{2k} x[n+2N/3] \right) W_N^{3kn} \qquad k = 0, 1, \cdots N/3-1$$

$$= \sum_{n=0}^{N/3-1} \left(x[n] + x[n+N/3] + x[n+2N/3] \right) W_{N/3}^{kn}$$

$$= \mathrm{DFT}\left(x[n] + x[n+N/3] + x[n+2N/3] \right) \qquad (N/3 \text{ point})$$

$$X[3k+1] = \sum_{n=0}^{N/3-1} \left(x[n] + W^{k+\frac{1}{3}} x[n+N/3] + W^{2k+\frac{2}{3}} x[n+2N/3] \right) W_N^{kn} \qquad k = 0, 1, \cdots N/3-1$$

$$= \sum_{n=0}^{N/3-1} \left(x[n] + \left(-\tfrac{1}{2} - \tfrac{\sqrt{3}}{2} j\right) x[n+N/3] + \left(-\tfrac{1}{2} + \tfrac{\sqrt{3}}{2} j\right) x[n+2N/3] \right) W_N^{(3k+1)n}$$

$$= \sum_{n=0}^{N/3-1} \left(x[n] + \left(-\tfrac{1}{2} - \tfrac{\sqrt{3}}{2} j\right) x[n+N/3] + \left(-\tfrac{1}{2} + \tfrac{\sqrt{3}}{2} j\right) x[n+2N/3] \right) W_{N/3}^{kn} W_N^{n}$$

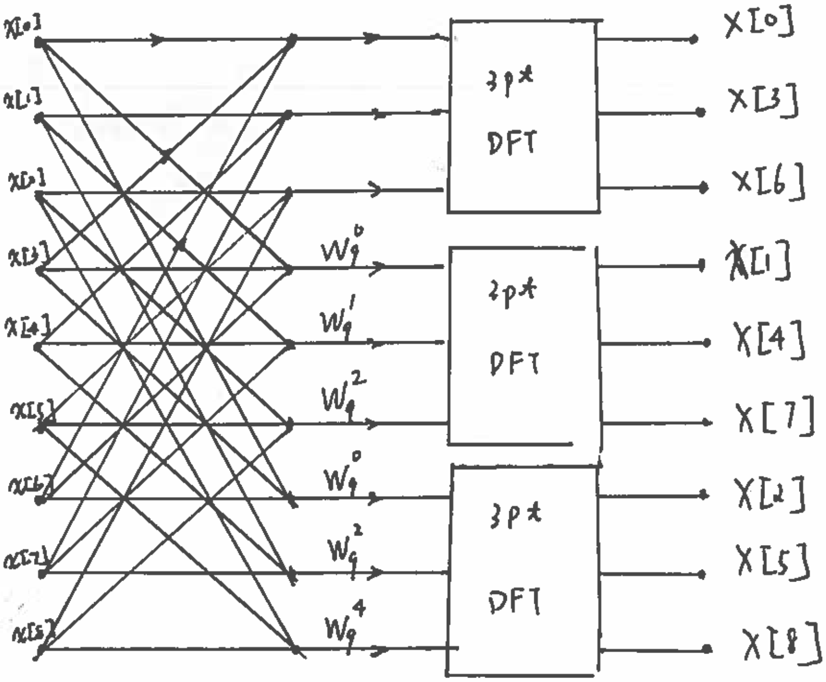
$$= \mathrm{DFT}\left(\left(x[n] + \left(-\tfrac{1}{2} - \tfrac{\sqrt{3}}{2} j\right) x[n+N/3] + \left(-\tfrac{1}{2} + \tfrac{\sqrt{3}}{2} j\right) x[n+2N/3] \right) W_N^{n} \right)$$

$$X[3k+2] = \sum_{n=0}^{N/3-1} \left(x[n] + W^{2/3} x[n+N/3] + W_N^{\frac{4}{3}} x[n+2N/3] \right) W_N^{kn} \qquad k = 0, 1, \cdots N/3-1$$

$$= \sum_{n=0}^{N/3-1} \left(x[n] + \left(-\tfrac{1}{2} + \tfrac{\sqrt{3}}{2} j\right) x[n+N/3] + \left(-\tfrac{1}{2} - \tfrac{\sqrt{3}}{2} j\right) x[n+2N/3] \right) W_N^{(3k+2)n}$$

$$= \mathrm{DFT}\left(\left(x[n] + \left(-\tfrac{1}{2} + \tfrac{\sqrt{3}}{2} j\right) x[n+N/3] + \left(-\tfrac{1}{2} - \tfrac{\sqrt{3}}{2} j\right) x[n+2N/3] \right) W_N^{2n} \right)$$

the diagram as follows. $N = 9$



in Butterfly

$x[n] + y[n] + z[n]$

$x[n] + (-\frac{1}{2} - \frac{\sqrt{3}}{2} j) y[n] + (-\frac{1}{2} + \frac{\sqrt{3}}{2} j) z[n]$

$x[n] + (-\frac{1}{2} + \frac{\sqrt{3}}{2} j) y[n] + (-\frac{1}{2} - \frac{\sqrt{3}}{2} j) z[n]$

Complexity of DFT: N^2 (multiple) $+ N^2$ (adding) $= 2N^2$

Complexity of 'radix-3 decimation-in frequency':

We have $\log_3 N$ DFT prosessor if N is keep divided by 3.

each preprocessor has 3 adding (if we store $(-\frac{1}{2} \pm \frac{\sqrt{3}}{2} j)$ as constant)

each preprocessor has 2 multiplying.

each DFT processor has 9 adding and 9 multiplying.

We have $N/3$ preprocessor and $\log_3 N$ DFT prosessor.

$\log_3 N$ Layer

the # of processor is $\frac{1 - 3^{\log_3 N}}{1-3} = \frac{N-1}{2}$

~~Complexity = $N/3 (3+2) (18 \log_3 N) \approx 36 N \log_3 N$~~

~~O('radix-3 decimation-in frequency') $= N \log_3 N$~~

$$\text{Complexity} = O(\text{Pre}) + O(\text{DFT})$$

$$= N/3 (3+2) \cancel{\tfrac{N}{2}} \log_3 N + (\frac{N-1}{2}) \log_3 N$$

$$= (\frac{5N}{3} + \frac{N-1}{2}) \log_3 N \approx O(\frac{5}{2} N \log_3 N)$$

3. (The Effect of zero-padding)

$$X[k] = \sum_{q=0}^{N-1} x[q]\,\delta[k-q]$$

$g_q[n] = \mathrm{IDFT}_N\{\delta[k-q]\}[n]$ $\tilde{g}_q[n]$ is the L-length zero-padding of $g_q[n]$

(a) Compute the closed-form expression for $\tilde{G}_q[k]$

Answer:
$$\tilde{G}_q[k] = \sum_{n=0}^{L} \tilde{g}_q[n]\, W_L^{kn}$$
$$= \sum_{n=0}^{N-1} g_q[n]\, W_L^{kn}$$

$$g_q[n] = \mathrm{IDFT}_N\{\delta[k-q]\}[n]$$
$$= \frac{1}{N}\sum_{k=0}^{N-1} \delta[k-q]\, W_N^{-kn}$$

$$\therefore \tilde{G}_q[k_2] = \sum_{n=0}^{NL-1} \left(\frac{1}{N}\sum_{k_1=0}^{N-1} \delta[k_1-q]\, W_N^{-k_1 n}\right) W_L^{k_2 n}$$
$$= \frac{1}{N}\sum_{k_1=0}^{N-1}\sum_{n=0}^{N-1} \delta[k_1-q]\, W_N^{-k_1 n}\, W_L^{k_2 n}$$
$$= \frac{1}{N}\sum_{k_1=0}^{N-1} \delta[k_1-q]\, \frac{1-W^{(-\frac{k_1}{N}+\frac{k_2}{L})N}}{1-W^{-\frac{k_1}{N}+\frac{k_2}{L}}}$$
$$= \frac{1}{N}\, \frac{1-W^{\frac{k_2 N}{L}}}{1-W^{-\frac{q}{N}+\frac{k_2}{L}}}$$

$$\therefore \tilde{G}_q[k] = \frac{1}{N}\, \frac{1-W^{\frac{kN}{L}}}{1-W^{-\frac{q}{N}+\frac{k}{L}}}$$

if $L = 2N$
$$\tilde{G}_q[k] = \frac{1}{N}\,\frac{1-W^{\frac{k}{2}}}{1-W_N^{-q+k/2}} = \frac{1}{N}\,\frac{W_4^{k}}{W_{2N}^{-q+k/2}}\,\frac{\sin(\frac{\pi k}{2})}{\sin(\pi\frac{-q+k/2}{N})}$$

$$\downarrow k \to 2(k+q)$$

$$G_q[2(k+q)] = \frac{1}{N}\,\frac{W_4^{k}}{W_{2N}^{k}}\,\frac{\sin(\pi(k+q))}{\sin(\pi\frac{k}{N})}$$

when $k = tN$, $t \in \mathbb{Z}$

$$G_q[2(k+q)] = \frac{W_4^{k}}{W_2^{t}}\, \cancel{(-1)^{q}\,\mathrm{sinc}(k\pi)}\, \delta(k-tN)$$

which is an effect of "periodic sinc interpolation"

3(b).
filename: solution3b.m

```
clear
x = double((imread('cameraman.tif')));
imgSize = size(x);
row = imgSize(1);
col = imgSize(2);
fX = zeros(imgSize);
for i = 1 : imgSize(1)
    fX(i,:) = fft(x(i,:));
end

%First way
newFx1 = zeros(imgSize(1),imgSize(2)+1024-256);
for i = 1 : imgSize(1)
    newFx1(i,:) = [fX(i,:),zeros(1,1024-256)];
end

%Second way
newFx2 = zeros(imgSize(1),imgSize(2)+1024-256);
for i = 1 : imgSize(1)
    newFx2(i,:) = [fX(i,1:col/2-1),zeros(1,1024-256),fX(i,col/2:col)];
end

%First way result
newImg1 = zeros(imgSize(1),imgSize(2)+1024-256);
for i = 1 : row
    newImg1(i,:) = ifft(newFx1(i,:));
end
%Second way result
newImg2 = zeros(imgSize(1),imgSize(2)+1024-256);
for i = 1 : row
    newImg2(i,:) = ifft(newFx2(i,:));
end
imagesc(x)
figure
imagesc(uint8(abs(newImg1)))
figure
imagesc(uint8(abs(newImg2)))
```

Result:

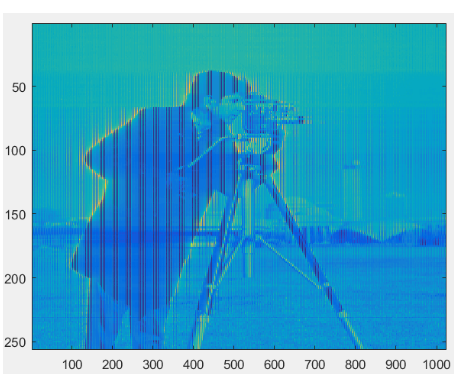


Figure 5: *Method1*

Figure 6: *Method2*

3(b) mathematical analysis.

first way: $\mathrm{IDFT}(\tilde{X}[k]) = \frac{1}{N}\sum_{k=0}^{L-1} X[k] W_L^{-nk}$ $x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k] W_N^{-nk}$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X[k] W_N^{-\frac{nkN}{L}}$$

$$\tilde{x}[n] = x[\tfrac{N}{L}n]$$

second way: $\mathrm{IDFT}(\tilde{x}[k]) = \frac{1}{N}\sum_{k=0}^{L-1} X[k] W_L^{-nk}$

$$= \frac{1}{N}\left(\sum_{k=0}^{N/2-1} X[k] W_L^{-nk} + \sum_{k=L-N/2}^{L-1} X[k-L+N] W_L^{-nk}\right)$$

$\downarrow k_2 = k-L+N$

$$= \frac{1}{N}\left(\sum_{k=0}^{N/2-1} X[k] W_L^{-nk} + \sum_{k_2=\frac{N}{2}}^{N-1} X[k_2] W_L^{-n(k_2+L-N)}\right)$$

$$= \frac{1}{N}\left(\sum_{k=0}^{N/2-1} X[k] W_L^{-nk} + \sum_{k_2=\frac{N}{2}}^{N-1} X[k] W_L^{-(k-N)n}\right)$$

$$= \frac{1}{N}\left(\sum_{k=0}^{N/2-1} X[k] W_L^{-nk} + \sum_{k=\frac{N}{2}}^{N-1} X[k] W_L^{-kn} \underline{\underline{W_L^{Nn}}}\right)$$

$\tilde{x}[n] = x[\frac{N}{L}n]$ when $\frac{Nn}{L}$ is integer.

$$\tilde{x}[n] = \frac{1}{N}\left(\sum_{k=0}^{N/2-1} X[k] W_L^{-nk} + \sum_{k=\frac{N}{N}}^{N-1} X[k] W_L^{-kn} W_L^{Nn}\right)$$

when $\frac{Nn}{L}$ is not integer.

therefore the second method interpolate some values.

4(a) 4(b)

```
x = double((imread('cameraman.tif')));
dctImage = dct2(x);
imagesc(uint8(log(abs(dctImage))));
colormap(gray);

energyTime = sum(x.^2);
energyDct = sum(dctImage.^2);
difference = energyDct - energyTime;
```

Result:
difference = 7.1526e-07

Figure 7: *Method1*

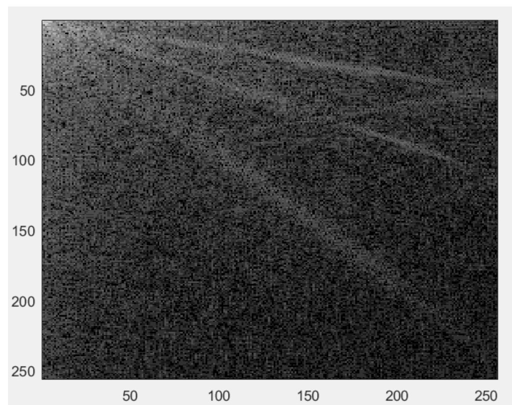


Figure 8: *Method2*

4(c)
A function called set zeros is set to do the set 0 processing.
filename: setZeros.m

```
function x = setZeros(x,alpha)
%x = setZeros(x,alpha)
%alpha:at what degree to set zeros
if alpha > 0
xSize= size(x);
xOp = sort(reshape(x,[1,xSize(1)*xSize(2)]));
position = round(length(xOp)*alpha);
valueForTest = xOp(position);
andGate = x > valueForTest;
x = x .* andGate;
end
```

filename: solution4c.m

```
x = double((imread('cameraman.tif')));
dctImageOrigin = dct2(x);
for i = 0.0 : 0.1 : 0.9
    figure
    dctImage = setZeros(dctImageOrigin,i);
    imageidct = uint8(idct2(dctImage));
```

```
    imagesc(imageidct);
    colormap(gray)
    title(num2str(i))
end
```

Results:

Figure 9: *0%*

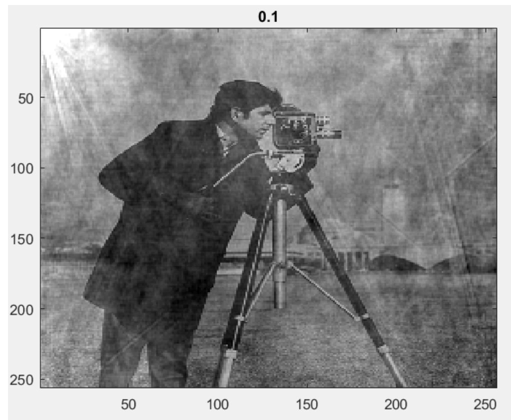


Figure 10: *10%*

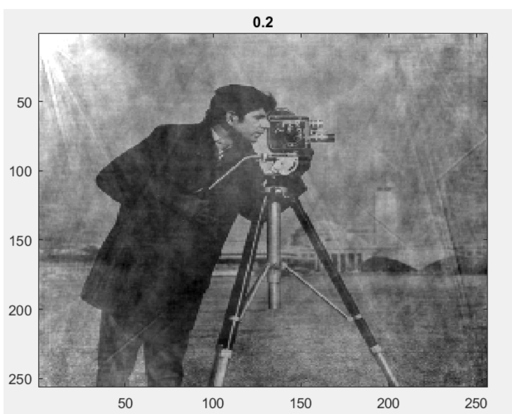


Figure 11: *20%*

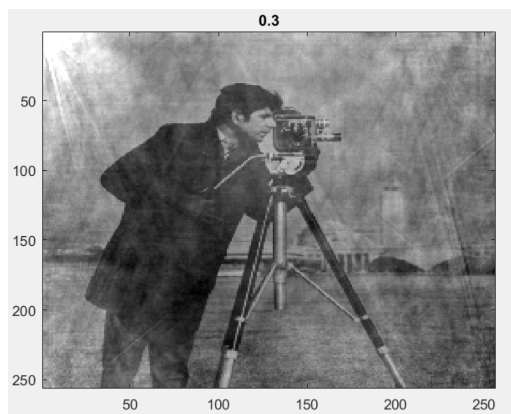


Figure 12: *30%*

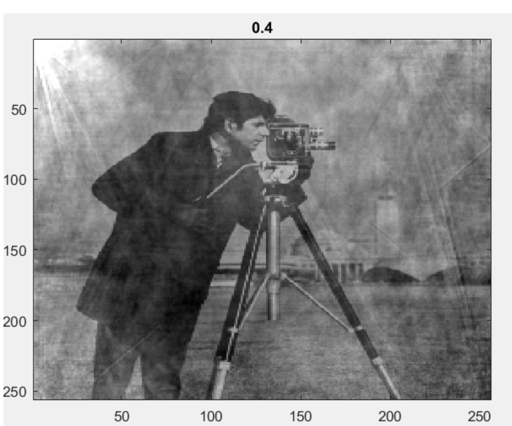


Figure 13: *40%*

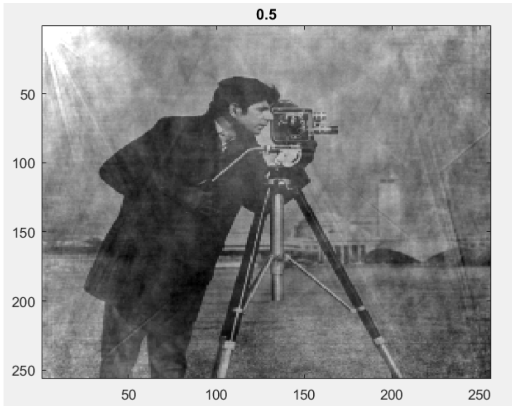


Figure 14: *50%*

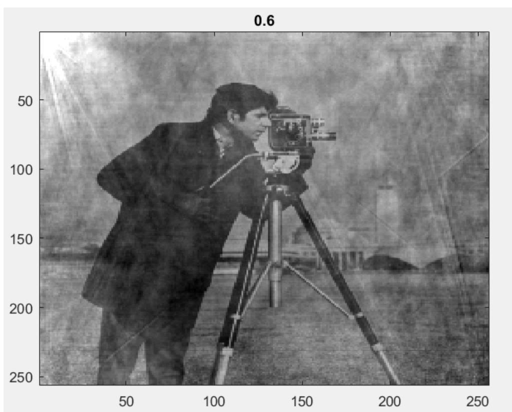


Figure 15: *60%*

Figure 16: *70%*

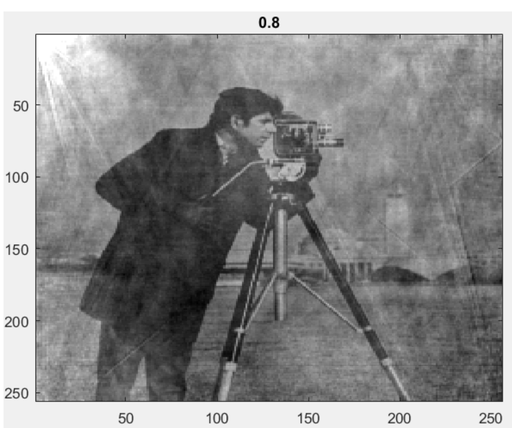


Figure 17: *80%*

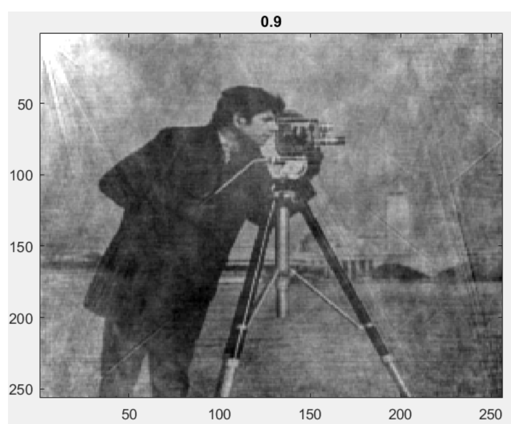


Figure 18: *90%*

4(d)
filename:solution4d.m

```
x = double((imread('cameraman.tif')));
tileDCT = x; %need the size
for i = 1 : 32
    for j = 1 : 32
        tile = x((i-1)*8+1:i*8,(j-1)*8+1:j*8);
        dctTile = dct2(tile);
        tileDCT((i-1)*8+1:i*8,(j-1)*8+1:j*8)=dctTile;
    end
end
imagesc((uint8(log(abs(tileDCT)))));
colormap(gray)
```

Result

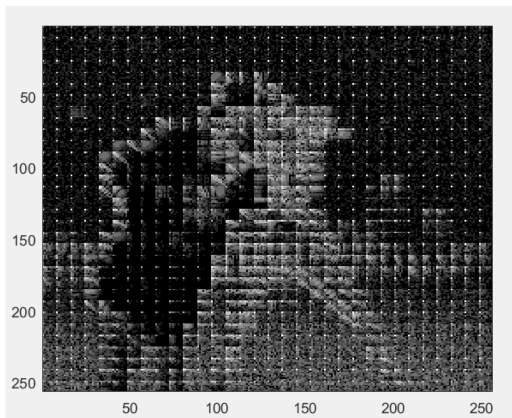


Figure 19: *Blocks of DCT*

5. (Haar Wavelet Transform) The Haar wavelet transform takes a length-N signal $x[n]$ for $0 \le n \le N-1$, and filters the signal with two different filers.

LPF: $h_1[n] = \{\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\}$. HPF $h_2[n] = \{-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\}$. Subsequently, the filtered signals are downsampled

$y_1[n] = x[n] \circledast h_1[n]$ $\quad y_2[n] = x[n] \circledast h_2[n]$. Haar wavelet coefficients $X[k]$ of the sequence $x[n]$ are defined as

$$X[k] = \begin{cases} y_1[2k], & k = 0, \dots, \frac{N}{2}-1 \\ y_2[2(k-\frac{N}{2})], & k = \frac{N}{2}, \dots, N-1 \end{cases}$$

$$= \begin{cases} \sum_{n=0}^{N-1} x[n] h_1[2k-n], & k = 0, \dots, \frac{N}{2}-1 \\ \sum_{n=0}^{N-1} x[n] h_2[2k-N-n], & k = \frac{N}{2}, \dots, N-1 \end{cases}$$

(a) Compute magnitude responses of $h_1[n]$ and $h_2[n]$, verify their type of filter.

$$X_1(e^{j\omega}) = \sum_{n=0}^{1} h_1[n] e^{-j\omega n} = \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} e^{-j\omega}$$

$$|X_1(e^{j\omega})|^2 = \left(\frac{1}{\sqrt{2}}\right)^2 (2 + 2\cos\omega + \cancel{\cos\omega}) = 1 + \cos\omega + \cancel{\cos\omega}$$

$\omega = 0$, $|X_1(e^{j\omega})|^2 = 2$

$\omega = \pi$, $|X_1(e^{j\omega})|^2 = 0$

$|X_1(e^{j\omega})|^2$ continous decreasing function over ω.

$\therefore$ $h_1[n] \rightarrow$ Low Pass filter.

$$X_2(e^{j\omega}) = \sum_{n=0}^{1} h_2[n] e^{-j\omega n} = -\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} e^{-j\omega}$$

$$|X_2(e^{j\omega})|^2 = \frac{1}{2}(2 - 2\cos\omega) = 1 - \cos\omega$$

$\omega = 0$ $\quad |X_2(e^{j\omega})|^2 = 0$

$\omega = \pi$ $\quad |X_2(e^{j\omega})|^2 = 2$

$|X_2(e^{j\omega})|^2$ continuous and increasing over ω.

$\therefore$ $h_2[n] \rightarrow$ High pass filter.

(b) Assuming N=6. Matrix representation of Haar wavelet transform.

$$X[k] = \begin{bmatrix} h_1[0] & h_1[-1] & \cdots & \cdots & h_1[1-N] \\ h_1[2] & h_1[1] & \cdots & \cdots & h_1[3-N] \\ \vdots & & & & \\ h_1[N-2] & h_1[N-3] & \cdots & & h_1[-1] \\ h_2[0] & h_2[-1] & \cdots & & h_2[1-N] \\ h_2[2] & h_2[1] & \cdots & & h_2[3-N] \\ \vdots & & & & \vdots \\ h_2[N-2] & h_2[N-3] & \cdots & & h_2[-1] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-2] \\ x[N-1] \end{bmatrix}$$

$$N=6 \quad , \quad X[k] = \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \end{bmatrix}}_{U_x} \cdot \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[5] \end{bmatrix}$$

(c) $U_x^H = U_x^T$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$U_x^H \cdot U_x = \begin{bmatrix} 1 & & & & & 0 \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ 0 & & & & & 0 \end{bmatrix} = \begin{bmatrix} E & 0 \\ 0 & 0 \end{bmatrix}$$

∴ it is "unitary" but with the last element missing.

d)
$$X[k] = U_x x[n]$$

$$U_x^{-1} X[k] = U_x^{-1} U_x x[n]$$

$$\therefore x[n] = U_x^{H} X[k] \qquad n = 0 \cdots \underbrace{N-2}_{}.$$

last element missing.