DIGITAL IMAGE PROCESSING 1

UE20EC317

Unit 2: Home Assignment Questions

SRN-PES1UG20EC083
NAME- Jacob V Sanoj

---

1. After converting to grayscale, perform discrete cosine transform on lena.png.
   (a) Display the result.
   (b) Set all values in the DCT matrix that are less than magnitude 50 to zero.
   (c) Now perform inverse discrete cosine transform on the image.
   (d) Using subplots, compare the two images and write your observations.
   (e) Also compare the quality in terms of Peak signal to noise ratio (PSNR)

Code

```
img = imread("lena.png");
L = rgb2gray(img);
W = dct2(L);
W(abs(W) < 50) = 0;
R = idct2(W);
R = rescale(R);
peaksnr = psnr(uint8(R), uint8(L));
disp(peaksnr)
montage({L,R})
```

OUTPUT
peaksnr = 6.873
Higher psnr give better quality image

2. Compute DST and IDST for Barbara image by writing a function to obtain the transformation kernel. Use subplots to compare the original image, the transformed image and the image obtained after performing IDST. (convert the image to grayscale and check if the rows and

columns are equal if not resize it) (**Use : ψ(m,n) =** $\sqrt{\frac{2}{N+1}}\frac{\sin\pi(k+1)(n+1)}{N+1}$; $\mathbf{0 \leq k, n \leq N-1,}$ **where ψ is the N x N sine transform matrix**)

CODE

```
PIC = imread("barbara.jpg");
K = rgb2gray(PIC);
K = im2double(K);

figure
subplot(1,3,1)
imshow(K)

J = dst(K);
K = J * K * J';
 subplot(1,3,2)
imshow(K)

L = J' * K * J;
subplot(1,3,3)
imshow(L)

function T = dst(img)
[x,~] = size(img);
N = x
T = zeros(N,N);
for k = 1:N
for n = 1:N
T(k,n) = sqrt(2/(N+1))*(sin((pi*(k+1)*(n+1))/(N+1)));
end
end
end
```

OUTPUT

3. After converting to grayscale, perform Discrete Sine Transform on lena.png.
(a) Display the result.
(b) Set all values in the DST matrix that are less than magnitude 0.5 to zero.
(c) Now perform inverse discrete sine transform on the image.
(d) Using subplots, compare the two images and write your observations.
(e) Also compare the quality in terms of Peak signal to noise ratio (PSNR)

CODE

```
b = imread("lena.png");
pic = rgb2gray(b);
pic = im2double(pic);

figure
subplot(1,3,1)
```

```
imshow(pic)

I = dst(pic)
c = I * pic * I';
subplot(1,3,2) imshow(K)


[N,~] = size(I);
mat = zeros(N,N);
for i = 1:N
for j = 1:N
if I(i,j) < 0.02
mat(i,j) = 0; else
mat(i,j) = I(i,j);
end
end
end

L = mat' * c * mat;
subplot(1,3,3)
imshow(L)

peaksnr = psnr(uint8(L), uint8(pic))
figure
montage({pic,L})

function T = dst(img)
[x,~] = size(img);
N = x
T = zeros(N,N); for k = 1:N
for n = 1:N
T(k,n) = sqrt(2/(N+1))*(sin((pi*(k+1)*(n+1))/(N+1)));
end
end
end
```

OUTPUT

Peaksnr = 48.732

The image that we obtained are not the same

4. Compute the Walsh Hadamard transform of Lena. Truncate values below 1. Reconstruct the image and compare the sizes.

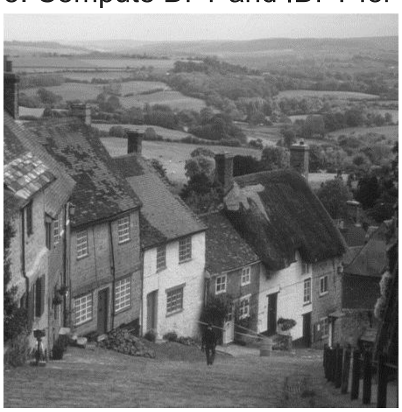
Code

```
v = imread("lena.bmp");
v = im2double(v);
t = fwht(v);
```

```
[N,~] = size(t);
for v = 1:N
for j = 1:N
r(v,j) = round(t(v,j),2);

end

figure
subplot(1,3,1)
imshow(t)

it = ifwht(t); subplot(1,3,2)
imshow(it)

rit = ifwht(r); subplot(1,3,3)
imshow(rit)
```

Output

5. Compute DFT and IDFT for the given image

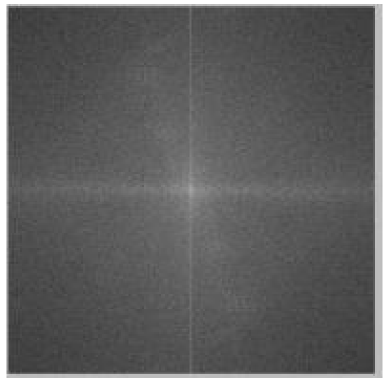


Code

```
pic = imread("assignment.png");
pic = rgb2gray(pic);
figure
subplot(1,3,1)
imshow(pic)
K = fft2(pic); D = abs(K);


fshift = fftshift(K);

fs = log(1+abs(fshift));
subplot(1,3,2)
imshow(fs,[])

ift = ifftshift(fshift);
f = ifft2(ift);
subplot(1,3,3)
```

```
imshow(f,[])
```

Output

PSNR tells us about the quality of the image,usually want higher psnr