Homework 6 Problem 7

EE 261: FT & Its Applications

Noor Fakih

Digital Watermarking

the image has been watermarked by:

- decomposed in 8 x 8 blocks
- computed the DCT of each block
- in each block, the bottom-right 2 by 2 block was erased and replaced by the amplitudes of the corresponding 2 by 2 block of the hidden image, divided by 10, making the dimensions of the hidden image equal to a quarter of the original image

Write a Matlab program that:

- reads the original image
- reconstructs the hidden image inside it
- use dct, idct, imread, and imwrite
- define the matrix D = dct(I_8) then the DCT of A is A_hat = DAD^T

```
% read in watermarked image
watermarked_img = imread("watermarked.png");

% Reconstruct Hidden Image
hiddenImg = recoverHiddenImg(watermarked_img);
imwrite(hiddenImg, 'hiddenImg1.png');

hiddenImg2 = recoverHiddenImg(hiddenImg);
imwrite(hiddenImg2, 'hiddenImg2.png', "png");
```

(a) The hidden image is an drawing of a man in pre-modern clothing.

(b) If you reiterate the process on the image discovered in part a, you will find Waldo!

Functions

```
function B = blockDCT(A)
% B is a matrix composed of the DCT of each 8 by 8 block
A_size = size(A);
D = dct(eye(8));
B = double(zeros(A_size));

for i=1:A_size(1)/8
    for j=1:A_size(2)/8
        A_block = double(A(8*(i-1)+1:8*i,8*(j-1)+1:8*j));
```

```
        B(8*(i-1)+1:8*i,8*(j-1)+1:8*j) = D*A_block*D';
    end
end
end

function hiddenImg = recoverHiddenImg(A)
% hiddenImg is the matrix that represents the image hidden
% inside the input matrix A
A_size = size(A);
A2 = blockDCT(A);
hiddenImg = zeros(A_size/4);
for i=1:A_size(1)/8
    for j=1:A_size(2)/8
        hiddenImg(2*(i-1)+1:2*i,2*(j-1)+1:2*j) = 10*A2(8*(i-1)+7:8*i,8*(j-1)+7:8*j);
    end
end

%% shift data to properly work with imwrite
% shift data so hiddenImg min is 0
hiddenImg = hiddenImg - min(hiddenImg(:));
% normalize the shifted data to 1
hiddenImg = hiddenImg / max(hiddenImg(:));

end
```