

Chapter 11.00D

Physical Problem for Fast Fourier Transform Computer Engineering

Problem Statement

Steganography is the art and science of hiding information in images. The basic idea is not to let the enemy know that message is being transmitted. This is different from message encryption, which is designed to make it hard to understand the message once a message is found. Steganography makes it hard to find the message itself. For instance, in Figure 1 one of the images has the preamble of the US constitution embedded in it. Can you guess? Looking at the images it is impossible to figure it out. There are various ways for performing steganography and there are various way to figure out if there is a hidden message in an image. Detecting steganography is another problem, which we will leave for another day. In this problem, we are going to look at one method for doing steganography using the Discrete Fourier Transform, computed using the Fast Fourier Transform.

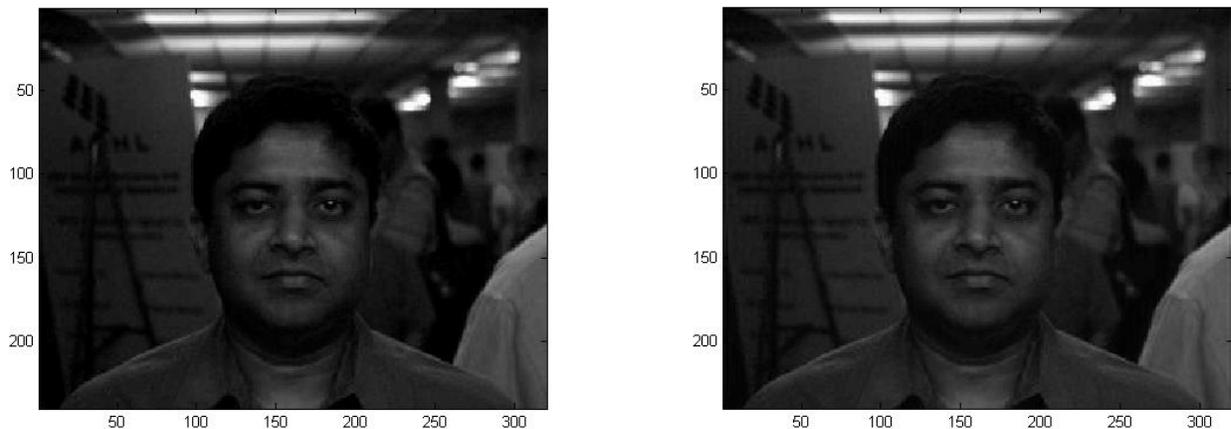


Figure 1 Which of these two images has a hidden message embedded in it? One of them has the amendments of the US constitution in it.

Problem Formulation

Image is a collection of gray level values at set of predetermined sites known as pixels, arranged in an array. These gray level values are also known as image intensities. For

this problem, an image is nothing but a two dimensional array of numbers, $I(m, n)$. The size of the array is the size of the image. One place to hide information in images is in the Fourier or frequency domain of images. Yes, we can define “frequency” of images too. Frequency in this context is defined with respect to the inverse of pixel or spatial units and not as inverse of time units. Imagine striped patterns, such as that one would see on shirts or on zebra. High frequency patterns would be closely spaced strips and low-frequency patterns would be those strips are that are widely spaced. Also, since images are two-dimensional quantities, the Fourier transform of an image is also two dimensional.

Since images are defined at discrete points, we will consider the discretized version of the Fourier transform. The discrete Fourier transform of an image (DFT) is given by

$$F(k, l) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(m, n) \exp\left(-i2\pi\left(\frac{mk}{M} + \frac{nl}{N}\right)\right)$$

Note the DFT is also a 2D array of the same size as the image, i.e. M by N . The inverse DFT (IDFT) is given by

$$I(m, n) = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F(k, l) \exp\left(i2\pi\left(\frac{km}{M} + \frac{ln}{N}\right)\right)$$

Figure 2 shows an example of the DFT, with just the magnitude visualized as images. There is, of course, the phase component too, which is not shown here. Note that the visualization might appear to be rather unconventional to you. Brighter regions correspond to larger magnitudes. As we see most of the information is towards the low-frequencies. So, it is not a good idea to change those. Higher frequencies control the sharpness of the image, so we cannot really mess around with that too. It is in the mid-range that we will store our information.

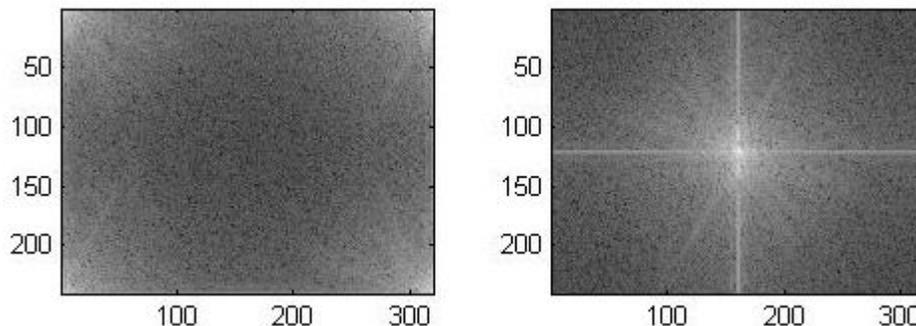


Figure 2 Discrete Fourier Transform of the face image show in Figure 1.

We show just the magnitude of the transform visualized as intensity. Whiter appearance corresponds to larger values. On the left is the unshifted version with the low-frequencies at the corners and on the right is the shifted version with the zero frequency at the center.

We will store our information by changing the magnitude of the DFT in an annular region, centered on the zero frequency information. We will simply replace the magnitude of the DFT in this region with corresponding ASCII value of the characters in the message. We will leave the phase information unchanged. Figure 3 shows examples of this hiding using

the text of the amendment of the US constitution. The MATLAB code for this encoding step is fairly simple.

```
F = fft2(double(I)); %Compute the FFT of the image
F2 = fftshift(F); %Shift the DFT so that the center is the low frequency
subplot(2,2,2); imagesc(log(abs(F2)));
[M N] = size(F); ci = round(M/2)+1; cj = round(N/2)+1;
lr = M/4; hr = M/4+50; %Dimension of the annular region for hiding
Mlen = length(Message); m=1;
for (i=1:hr)
    for (j=1:hr)
        r = sqrt(i*i + j*j);
        if ((r > lr) && (r < hr))
            if (m <= Mlen)
                F2(i+ci, j+cj) = double(Message(m));
                F2(i+ci, -j+cj) = F2(i+ci, j+cj); % copy over to each
                quadrant
                F2(-i+ci, j+cj) = F2(i+ci, j+cj);
                F2(-i+ci, -j+cj) = F2(i+ci, j+cj);
                m = m+1; end; end; end; end;
```

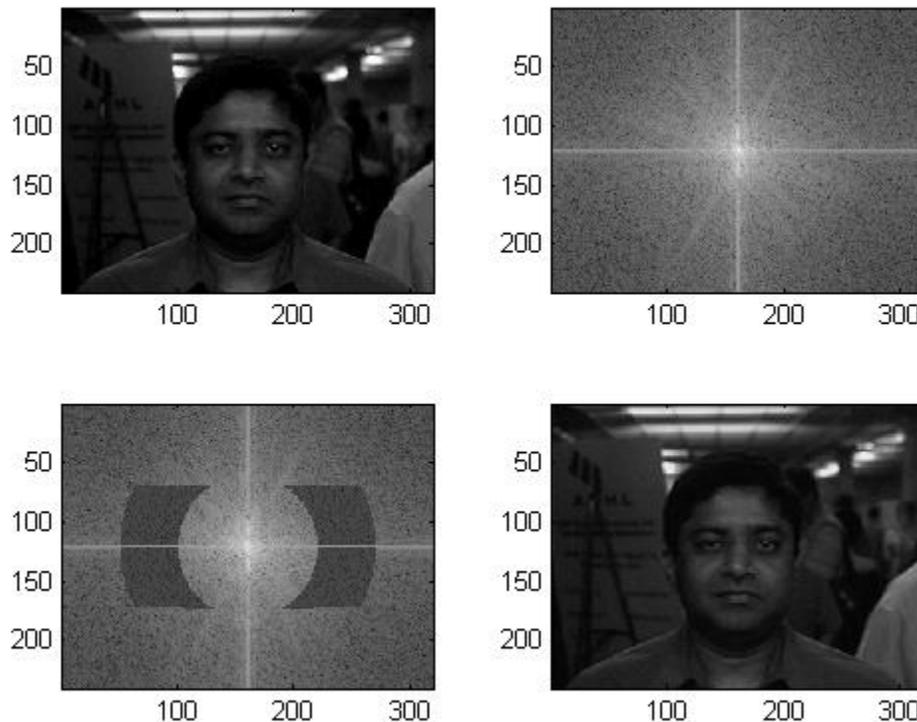


Figure 3 Top row shows the original image along with (shifted, centered) magnitude of the DFT. In the bottom row, we have the DFT magnitude image, with the message embed in it. The portion where the message is stored is obviously the darker annular pieces. The image on

the right of the bottom row is the reconstructed image. Note from the reconstructed image, it is hard to make out if any message has been stored in it.

To retrieve the message, we simply have to perform the DFT (or rather FFT) of the received images, and just read of the magnitude in the annular region, whose location has been decided upon at an earlier stage, through different channels. The retrieve step is also fairly simple. Here is a MATLAB code snippet:

```
% Retrieve
F = fft2(J); F2 = fftshift(F); Mlen = length(Message); m = 1;
for (i=1:hr)
    for (j=1:hr)
        r = sqrt(i*i + j*j);
        if ((r > lr) && (r < hr))
            if (m <= Mlen)
                fprintf(1, '%c', char(uint8(real(F2(i+ci, j+cj)))));
                m = m + 1;    end;    end;    end; end;
```

Questions and Assignments

- Try storing information in different portions of the DFT space and see what happens to the image.
- What happens if we hide the information in the phase? How about just changing the real or the imaginary part of the DFT coefficients?
- As fun, try exchanging the magnitude and the phase information in the DFTs of two images and then reconstructing them (inverse-DFT). You will be surprised at how much information is stored in the DFT phase of the image.

Topic	FAST FOURIER TRANSFORM
Sub Topic	Physical Problem
Summary	Steganography: Hiding secret messages in images!
Authors	Sudeep Sarkar
Last Revised	August 16, 2009
Web Site	http://numericalmethods.eng.usf.edu