Solutions For Homework #4

Problem 1:[10 pts]

The function $f(x, y)$ shown in the Figure can be expressed mathematically as follows

$$f(x, y) = \text{rect}\left(\frac{\sqrt{(x-3)^2 + (y-2)^2}}{2}\right) - \text{rect}(\sqrt{(x-3)^2 + (y-2)^2})$$

where $\text{rect}(\sqrt{x^2 + y^2})$ is a function that is equal to unity over a central circle of unit *diameter*, i.e.

$$\text{rect}(r) = \{ \begin{array}{c} 1, \quad r = \sqrt{x^2 + y^2} < 0.5 \\ 0 \quad \text{else} \end{array}$$

Now, the circular rect function $\text{rect}(r)$ forms a Fourier Transform pair with the jinc function,

$$\text{rect}(r) \supset \text{jinc}(q) = \frac{J_1(\pi q)}{2q}, \quad q = \sqrt{u^2 + v^2}$$

Note that the independent variables r and q above are, respectively, distance and spatial frequency quantities in polar coordinates. By the Scaling Theorem (see problem 2),

$$\text{rect}(Dr) \supset \frac{1}{D^2}\text{jinc}\left(\frac{q}{D}\right)$$

and by the Shift theorem,

$$\text{rect}(\sqrt{(x-x_0)^2 + (y-y_0)^2}) \supset \text{jinc}(q)e^{-j2\pi(ux_0 + vy_0)}$$

Putting all these results together, we can find the Fourier Transform of the function $f(x, y)$ given in Equation

$$\begin{aligned} \text{rect}\left(\frac{\sqrt{(x-3)^2 + (y-2)^2}}{2}\right) &\supset 4\text{jinc}(2q)e^{-j2\pi(3u+2v)} \quad (1) \\ \text{rect}(\sqrt{(x-3)^2 + (y-2)^2}) &\supset \text{jinc}(q)e^{-j2\pi(3u+2v)} \\ \Rightarrow f(x, y) &\supset e^{-j2\pi(3u+2v)}[4\text{jinc}(2q) - \text{jinc}(q)] \end{aligned}$$

Problem 2:[10 pts]

We know that $f(x,y)$ and $F(u,v)$ form a Fourier Transform pair, i.e.

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-j2\pi(ux+vy)}\ dx\ dy$$

Now, consider the integral

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(ax,by)e^{-j2\pi(ux+vy)}\ dx\ dy$$

We make a change of variables

$$\begin{aligned}\tilde{x} &= ax \\ \tilde{y} &= by\end{aligned} \qquad (2)$$

The Jacobian matrix of this linear transformation is

$$J = \begin{bmatrix} \frac{\partial\tilde{x}}{\partial x} & \frac{\partial\tilde{x}}{\partial y} \\ \frac{\partial\tilde{y}}{\partial x} & \frac{\partial\tilde{y}}{\partial y}\end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b\end{bmatrix}$$

and the new area element $d\tilde{x}d\tilde{y} = |J|dxdy = ab\ dxdy$. Thus, with this change of variables introduced, Equation becomes

$$G(u,v) = \frac{1}{|a||b|}\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tilde{x},\tilde{y})e^{-j2\pi(\frac{u}{a}\tilde{x}+\frac{v}{b}\tilde{y})}\ d\tilde{x}\ d\tilde{y}\right]$$

But, the integral in square brackets can be recognized as the Fourier transform of $f(x,y)$ evaluated at $\frac{u}{a}$ and $\frac{v}{b}$. So,

$$f(ax,by) \supset G(u,v) = \frac{1}{|a||b|}F(\frac{u}{a},\frac{v}{b})$$

Problem 3:[10 pts]

In this problem, the column direction is assigned the x axis, while the row direction is called y. The same designation applies to the frequency axes, i.e. the

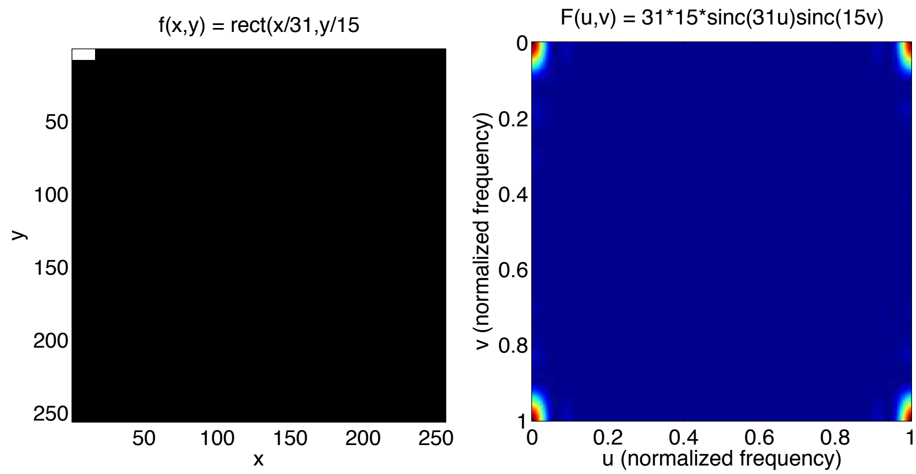

Figure 1:

columns refer to the u spatial frequency axis while the rows denote the v axis. We display computed spectra with the DC value located in the upper left hand corner of the matrix. Thus, the spatial frequencies run from 0 to 1, in normalized frequency units, in both column and row directions.

(a) The function $f(x,y) = \text{rect}(x/31, y/15) \quad 0 <= x, y <= 255$ is shown in Figure 1. The magnitude spectrum, i.e. the absolute value of the Fourier Transform of $f(x,y)$, is also shown in Figure 1. Analytically, we would expect the following Fourier relationship

$$f(x,y) = \text{rect}(x/31, y/15) \supset F(u,v) = 15 \times 31 \times \text{sinc}(31u)\text{sinc}(15v)$$

From this relationship, we would expect the width of the spectrum along the u (column) axis to be about $\frac{1}{15} = 0.0667$ and the width along the v row axis approximately $\frac{1}{7} = 0.14$ in normalized frequency units. Indeed, as can be seen on the left side in Figure 3, which is a zoomed-in version of the upper left-hand corner of Figure 1, we observe that the first null of the sinc function in the column (u) direction occurs at about 0.06 whole the first null in the row (v) occurs at about 0.15, consistent with what we expect.

(b) The function $f(x,y) = \text{rect}(r/15) \quad -128 <= x, y <= 127$ is shown in Figure 2. The magnitude spectrum, i.e. the absolute value of the Fourier Transform of $f(x,y)$, is also shown in Figure 2. Analytically, we would

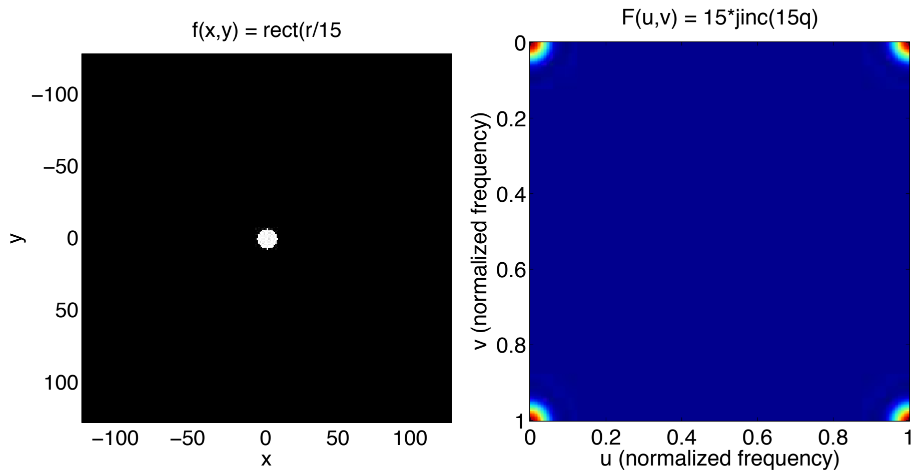

Figure 2:

expect the following Fourier relationship

$$f(x,y) = \text{rect}(r/15) \supset F(u,v) = 15^2 \times \text{jinc}(15q)$$

where, $r = \sqrt{x^2+y^2}$ and $q = \sqrt{u^2+v^2}$. From this relationship, we would expect a circularly symmetric spectra, as evidenced by Figure 2. Furthermore, we expect the width of the spectrum, *in the radial direction*, to be about $\frac{1}{7} = 0.14$. By inspection of the right side of Figure 3, which is a zoomed-in version of the upper left hand corner of Figure 2, we see that the first null of the jinc function occurs when $u \approx 0.9$ and $v \approx 0.9$, giving a radial distance of about $q \approx \sqrt{2} \times 0.9 = 1.3$, which is consistent with our expectations.

Problem 4:[10 pts]

Recall the Shift Theorem, which says

$$f(x-x_0, y-y_0) \supset F(u,v)e^{-j2\pi(ux_0+vy_0)}$$

where $f(x,y)$ and $F(u,v)$ form a Fourier Transform pair. In words, this Theorem states that a translation of a 2D function in the spatial domain results in multiplication of the function's spectrum by a phase ramp in the frequency domain. In

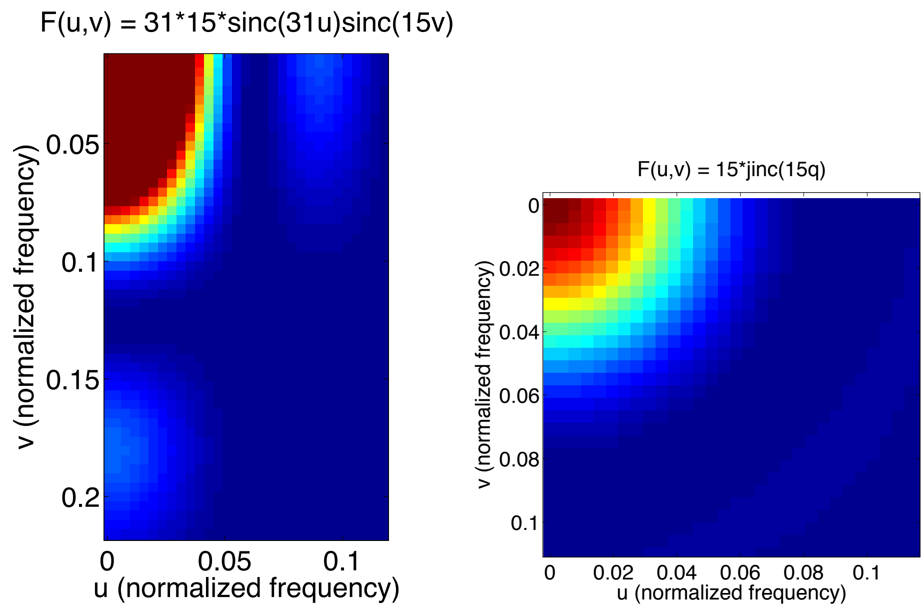

Figure 3:

this problem, we will make use of the discrete version of the Shift Theorem for Fourier Transforms to implement the desired image shift of 27.3 pixels to the right and 35.2 pixels in the downward direction.

When the shifts (x_0, y_0) are integers, the phasor in Equation is easily constructed as follows

$$e^{j2\pi(\frac{k}{N}x_0+\frac{l}{M}y_0)} \quad k = 0, ..., N-1,\ l = 0, ..., M-1$$

where, for N horizontal frequency bins and M vertical frequency bins, the discrete frequencies are $u = \frac{k}{N}$ and $v = \frac{l}{M}$. We note, in particular, that

$$\begin{aligned} e^{j2\pi\frac{k}{N}x_0} &= e^{-j2\pi\frac{N-k}{N}x_0} \\ e^{j2\pi\frac{l}{M}y_0} &= e^{-j2\pi\frac{M-l}{M}y_0} \end{aligned} \tag{3}$$

when x_0 and y_0 are integers. Recall that, for a real-valued discrete time signal $f(n)$ (let's take the 1D case for simplicity),

$$f(n) \supset F(k) \text{ and } F(N-k) = F^*(k)$$

where $*$ denotes complex conjugation. The same applies for two dimensional signals. Thus, when x_0 and y_0 are integers, multiplying the 2D spectrum of an image by a phasor, constructed above, and then taking the inverse Fourier Transform results in a real-valued image shifted by x_0 pixels across and y_0 pixels down. The problem arises when the fractional shifts, as is our case, are desired. In this case, we need construct the phasor to explicitly obey the constraint in (4) above. The MATLAB code for doing this is given at the back. The original image is shown in Figure 4 while the shifted image is given in Figure 5. Note how the shift, for

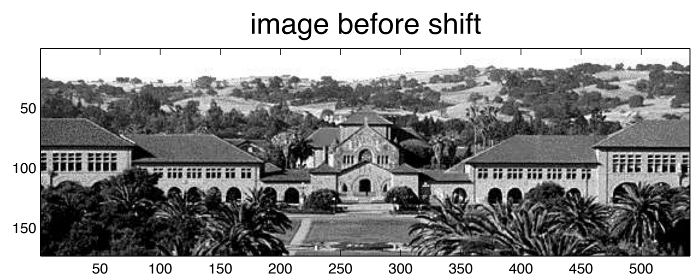

Figure 4:

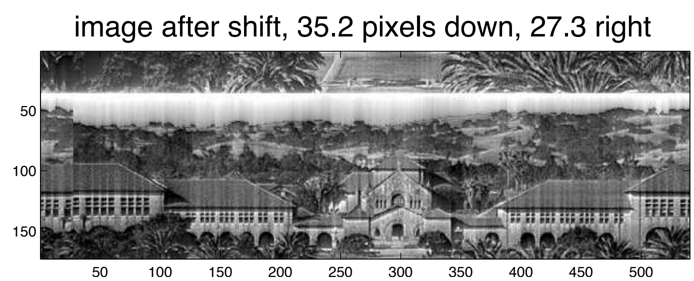

Figure 5:

finite-length signals (or images in our case), actually wraps the image around the edges, i.e. the shift in both the column and row directions is circular.

Problem 5:[10 pts]

(a) The amplitude image from the data file *hw4prob5data* required a linear stretch before display. We applied the linear stretch equation from Homework 1

$$\text{image}_{stretched} = 80\frac{\text{image} - \mu}{\sigma} + 128 \quad (4)$$

where $\mu = 59$ and $\sigma = 81.5$ refer to the mean and standard deviation of pixel values in the original image. The result after stretching is shown in Figure 6 We note that the width of the box in the center of the image is

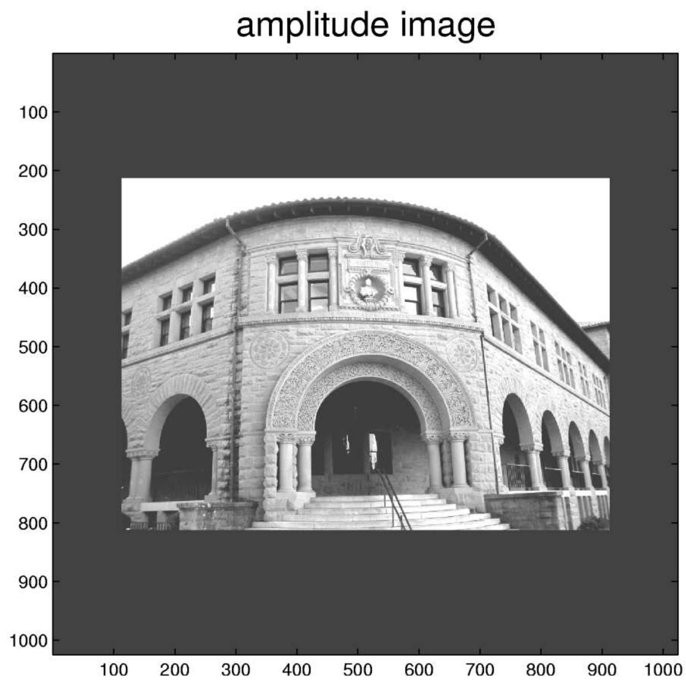

Figure 6:

about 800 pixels, its height about 600 pixels.

(b) We compute the 2D Fourier Transform of the image using the FFT2 routine in MATLAB. Displayed in Figure 7 is the *magnitude* of the spectrum, in DB, using the formula $20log_{10}(spectrum)$. We note that the u and v frequency axes have been set such that DC is located in the center of the matrix. Furthermore, the u and v axes are in normalized frequency units.

(c) Figure 7 shows that most of the energy is concentrated near the spatial frequencies $u \approx -0.23$ and $v \approx 0.17$. It is as if the entire frequency content has been shifted from DC to those coordinates. By the Shift Theorem, we know that

$$F(u - u_0, v - v_0) \supset f(x, y)e^{j2\pi(xu_0 + yv_0)} \quad (5)$$

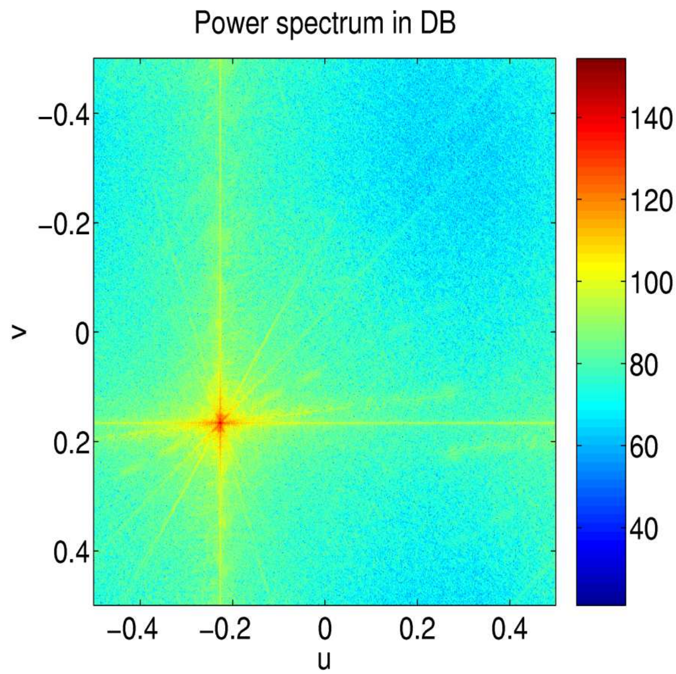

Figure 7:

In other words, a translation of the origin in the frequency domain (in the case above, from $(0, 0)$ to about $(-0.23, 0.17)$) results in a phase modulation in the space domain. Indeed, by looking at the angle of the complex numbers comprising the image formed from *hw4prob5data*, we see the modulation pattern shown in Figure 8: Notice that the wave-like behavior of the modulation is a consequence of the phase of $e^{j2\pi(xu_0+yv_0)}$. The direction perpendicular to the phase fronts exactly equals to the translation direction of the spectrum in the Fourier domain.

Figure 8 shows a zoomed in version of the translated spectrum. As mentioned previously, most of the energy seems to be concentrated near $u \approx -0.23$ and $v \approx 0.17$. Thus, neglecting the phase modulation, the image is generally smooth. Furthermore, we observe a certain symmetry in the 2D distribution of Fourier coefficients in Figure 7. This is due to the real-valued nature of the image. It can be seen, from Figure 8, that most of the enery is located in a square of width about 0.08 normalized frequency units in length. This "square" can be considered the effective bandwidth of the im-

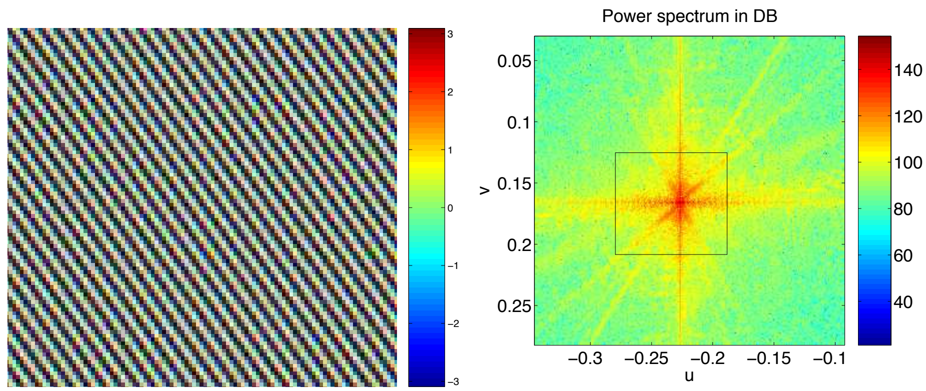

Figure 8:

age. From Figure 8, we also observe energy distributed in frequency bins on lines oriented at an angle to the u axis. This directed distribution of energy implies that 2D sinusoidal variations in the same direction are particularly pronounced in the image.

Problem 6:[10 pts]

(a) Figure 9 shows the spectrum magnitude computed using MATLAB's FFT subroutines. The reported elapsed computation time was 0.1869 seconds.

(b) Our implementation of the DFT algorithm involves matrix-vector multiplication. Recall that the DFT for a N-point signal (i.e. sequence of numbers), $f(n)$, is computed as follows

$$F(k) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi \frac{nk}{N}} \quad k = 0, ..., N-1 \tag{6}$$

Note that the result of the DFT is a new sequence, $F(k)$, of length N representing the N Fourier coefficients. We also observe that the equation above

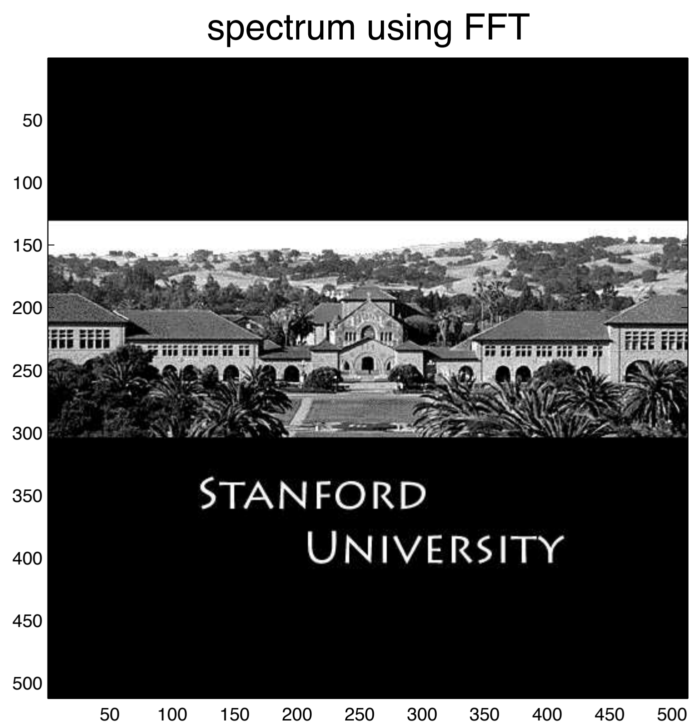

Figure 9:

can be implemented as a dot product between two vectors

$$F(k) = [f(0)\ f(1)\ ...\ f(N-1)] \begin{bmatrix} 1 \\ e^{-j2\pi\frac{k}{N}} \\ . \\ . \\ . \\ e^{-j2\pi\frac{(N-1)k}{N}} \end{bmatrix} \tag{7}$$

The result of the dot product above is the k-th DFT coefficient. Hence, we can form the entire N-length sequence of DFT coefficients by multiplying the signal $f(n)$ with a matrix A whose ij-th entry is

$$A_{ij} = e^{\jmath 2\pi\frac{ij}{N}} \tag{8}$$

This is how we implement our DFT algorithm. In particular, we loop down the rows of the image, computing a 1D DFT for each line and storing the result. Subsequently, we loop across the columns and compute 1D DFTs down the rows. We find that the running time for our MATLAB implementation of the DFT is about 4.13 seconds. We see that our the FFT implementation of the DFT performs much faster than our implementation of the DFT.

The way we have implemented our DFT algorithm, we would expect there to be $MN^2 + NM^2$ operations for an $M \times N$ image. We get this by noting that a matrix-vector multiplication for a row of length N, for example, requires N^2 operations and there are M rows, giving a total of MN^2 operations. On the other hand, we know that a 1D FFT operation on a row of length N costs $N \log N$ operations. With M rows, the cost would be about $MN \log N$, giving a total (FFTs across the rows and columns) of $MN \log MN$. Thus, the theoretical savings achieved by using FFT routines as opposed to our DFT implementations would be of the order $\frac{MN \log MN}{MN^2+NM^2} = \frac{\log MN}{N+M}$. The resulting image is shown in Figure 10

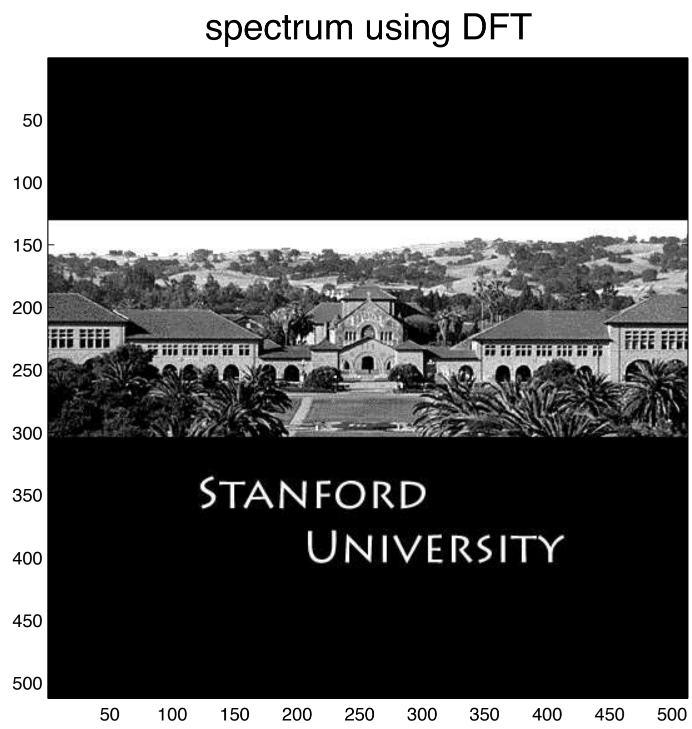

Figure 10:

MATLAB code for Problem 4

```
% EE 262 Spring 2005, Problem Set 4
%----------------------------------%

% Problem 4
%-----------%

% load in file
fid = fopen('stanfordbw','rb');
 data = fread(fid, inf, 'uint8');fclose(fid)

% reshape into image
im = reshape(data, [540 173]);im = im';

% note: we define u along the columns, and v DOWN the rows

% create a grid of (u,v) frequencies, of dimensions
% 173 rows 560 columns, with u = 0, 1/540,...,539/540,
%  v = 0,1/173,...,172/173

[u,v] = meshgrid([0:540-1]/540,[0:173-1]/173);

% create phase ramp in column direction (right)
phase_rampu = zeros(size(im));
phase_rampu(:,1:540/2) = 2*pi*(u(:,1:540/2)*27.3);
phase_rampu(:,540:-1:540/2+1) = -2*pi*(u(:,1:540/2)*27.3);

% create phase ramp in row direction (down)
phase_rampv = zeros(size(im));
phase_rampv(1:(173+1)/2,:) = 2*pi*(v(1:(173+1)/2,:)*35.2);
phase_rampv(173:-1:(173+1)/2+1,:)=-2*pi*(v(1:(173+1)/2-1,:)*35.2);
```

```
% apply phase ramp to image spectrum and inverse transform

exp_phase_ramp=exp(-sqrt(-1)*phase_rampu).*exp(-sqrt(-1)*phase_rampv);
im_shifted = ifft2( fft2(im).*exp_phase_ramp);

% display

figure(1); imagesc(im);axis image; caxis([0 255]);colormap bone;
h1=title('image before shift');set(h1,'FontSize',20);

figure(2); imagesc(real(im_shifted));axis image;
 caxis([0 255]);colormap bone;
h1=title('image after shift, 35.2 pixels down, 27.3 right ');
set(h1,'FontSize',20);
```

MATLAB code for Problem 6

```
% EE 262 Spring 2005, Problem Set 4
%-----------------------------------%

% Problem 6
%-----------%

% part (a)

% load in file
fid = fopen('hw4prob6data','rb');
data = fread(fid, inf, 'float');fclose(fid)

% reshape into image
im = reshape(data, [512*2 512]);im = im';

im = im(:,1:2:end-1) + sqrt(-1)*im(:,2:2:end);


% part (a)
```

```
% compute spectrum using FFT
tic
spec1 = fft2(im);
toc

% part (b)

% compute spectrum using DFT

% form DFT matrix, n increases along columns,
%  k down rows

[n,k] = meshgrid([0:512-1],[0:512-1]);
DFT = exp(sqrt(-1)*2*pi*(n).*(k)./512);
spec2 = zeros(512,512); spec3 = zeros(512,512);
tic
% DFT in column direction
for rows = 1:512
  x = im(rows,:)';
  spec2(rows,:) = ( (DFT*x)' );
end

% DFT in row direction
for cols = 1:512
  x = spec2(:,cols);
  spec3(:,cols) = ( (DFT'*x) );
end
toc

% display
figure(1); imagesc(abs(spec1));axis image;colormap bone;
h1=title('spectrum using FFT'); set(h1,'FontSize',20);
figure(2); imagesc(abs(spec3));axis image;colormap bone;
h1=title('spectrum using DFT'); set(h1,'FontSize',20);
```