Activity 2. Fourier Transform Model of Image Formation (Part 1 of 2)

Definitions

The Fourier transform (FT) transform is one among the many classes of linear transforms that recast information into a set of linear basis functions. In particular, FT converts a signal with physical dimension x into a signal with dimension 1/x using sinusoids as basis functions. If the signal has dimensions of space, the FT of the signal will have dimensions of inverse space or spatial frequency.

The FT on a two-dimensional signal f(x,y) is given by

$$F(f_x, f_y) = \int\int f(x, y) exp(-i2\pi(f_x x + f_y y)) dx dy \tag{1}$$

where fx and fy are the spatial frequencies along x and y respectively.

An efficient and fast implementation of FT is the Fast Fourier transform algorithm or FFT developed by Cooley and Tukey [1]. All scientific programming language have FFT routines. In fact, because of its usefulness FFT is already embedded in the hardware of most digital devices such as PC's, tablets and mobile phones.

The 2D FFT has the following properties:

1. The output of FFT2 has quadrants along the diagonals interchanged. That is if the quadrants are labeled in the clockwise direction as $\begin{matrix} 4 & 1 \\ 3 & 2 \end{matrix}$ the FFT output comes out as $\begin{matrix} 2 & 4 \\ 1 & 3 \end{matrix}$. The function fftshit () interchanges the quadrants back.
2. The output of the FFT2 is a **<u>complex number array</u>**. To view the intensity use abs() function which computes the modulus of the a complex number.
3. The inverse FFT2 , ifft2( ), is just the same as the forward fft2( ) only that the output image may be upside down.

FFT allows us to compute the FT of a function for which an analytic solution is difficult to derive. The following activities demonstrate the properties of FFT and its applications.

Activity 2.1 Familiarization with Discrete FT

1. Create an image of a white circle against a black background centered in a 400x400 pixel matrix. Suppose the image is called A. You may use the image you used in Activity 1.
2. Apply fft2() on the image and compute the intensity values using abs(). Remember, the result of and FFT is a complex matrix. Display the FFT magnitude as an intensity image.

MATLAB	PYTHON
FA = fft2(A); figure (1); image(abs(A));	FA = np.fft.fft2(A) plt.imshow(abs(FA), cmap = 'gray')

3. Notice that the resulting image have intensities at the corners. This is because of the property of FFT2 where the diagonal quadrants are interchanged. To make the FFT2 output appear zero-centered, use fftshift and then display. Use the "hot" colormap to make it look like a laser diffraction pattern.

MATLAB	PYTHON
FAshifted = fftshift(FA); figure (2); image(abs(FAshifted)); colormap hot;	FAshifted = np.fft.fftshift(FA) plt.imshow(abs(FAshifted), cmap = 'hot')

4. Pick any image in your collection and crop out any square portion. Load this image in your program and convert this image into grayscale. Apply fft2 () twice on the image and display the reconstruction. Do it again, but this time apply fft2() followed by ifft2(). Comment on the appearance of the reconstructed images.

MATLAB	PYTHON

I = rgb2gray(imread('mypicture.jpg'));
figure(3);
subplot(1,3,1); imshow(I);
Irec = fft2 (fft2(I));
subplot(1,3,2); imagesc(abs(Irec));
colormap gray; axis image;
Irec2 = ifft2(fft2(I));
subplot(1,3,3);
imagesc(abs(Irec2)); colormap gray; axis image;
	(Fill in the blanks)

Convolution

The convolution between two 2-D functions f and g is given by

$$h(x, y) = f \circledast g = \int\int f(x', y')g(x - x', y - y')dx'dy'. \tag{2}$$

The convolution is a linear operator which means that if f and g are recast by linear transformations such as the Laplace or Fourier transform, they obey the convolution theorem,

$$H = FG \tag{3}$$

Where H, F and G are the transforms of h, f and g respectively. This means, a convolution in 2-D space is a multiplication in Fourier space.

The convolution is a "smearing" of one function against another, such that the resulting function h looks a little like both f and g. Convolution is used to model the linear regime of instruments or detection devices such as in imaging. For example, f can be the object, and g can be the impulse response of the imaging system. Their convolution, h, is then the image produced by the detection system which as you will see is not perfectly identical to the original object.

For imaging systems that use a lens with a circular aperture the transfer function is due to the finite size of the camera lens and aperture. A smaller lens radius or aperture means the lens can only gather a small bundle of rays that reflect off an object, therefore the image of the object is never perfect. However, as the aperture or lens diameter increases, more details of the object can be seen in the image.

Activity 2.2. Simulation of an imaging system

In optics, an aperture that is lit by coherent monochromatic light (e.g. laser) will produce a diffraction pattern at a far distance known as the Fraunhofer diffraction pattern and the pattern is the Fourier transform of the aperture. The image in Activity 1 is the Airy pattern, the Fraunhofer diffraction pattern of a circular aperture. The FT of the aperture is also its **point spread function (PSF)** which shows how an image of a point object will appear in the imaging system. Each pixel in a pristine (ideal, sharp) image can be considered a point object. Thus, the PSF of an imaging system convolved with a pristine image results in the image produced by that imaging system.

1. Create a 256x256 image of the letters "**NIP**" in large bold fonts (Arial or Helvetica font recommended) using PAINT or similar apps. Let the letters fill 50% to 75% of the space. Save this image as TIF to preserve the crisp edges. Note the edges of the sans serif font are sharp.
2. Create another 256 x 256 image of a white circle (centered) against a black background. This image represents the "aperture" of a circular lens. Let the diameter of this circle be at 10% of the image width.
3. Convolve the two images using the following steps :
   3.1. FFTshift the circular aperture. For a lens, this aperture function is already in the FT space so no need to FFT2 the aperture.
   3.2. Take the FFT2 of the NIP image. Do not use abs. We need both real and imaginary parts.
   3.3. Multiply the FFT2(NIP) with the fftshifted aperture.
   3.4. Inverse FFT2 the product in 3.3 using ifft2();
   3.5. Plot the modulus (abs ()) of the result in 3.4.
4. Repeat steps 2 and 3 for apertures with diameters 25%, 50%, 75% and 100% of the array width. Comment on the appearance of the reconstruction.

See the sample code below:

MATLAB	PYTHON

Ashift = fftshift(A);
%when you load the image it might be in RGB.
%Just pick one channel or else convert the image into gray first.
FImg = fft2(I(:,:,1));
H = Ashift.*FImg;
h = ifft2(H);
figure(4);
imagesc(abs(h)); colormap gray; axis image;
	Ashift = np.fft.fftshift(A) # A is the circular aperture
	FImg = np.fft.fft2(Img[:,:,1])
	# If you load the image it might be in RGB channels, just pick # one channel.
	H = Ashift*FImg
	h = np.fft.ifft2(H)
	plt.imshow(abs(h), cmap='gray') Fill in the blanks)

5. Simulate the image of a star produced by the James Webb Space Telescope. Stars may be considered point objects. Simply get the FT of the JWST honeycomb mirror configuration you created in Activity 1. The resulting FT image is how a single star would look like as imaged by JWST.

Correlation

The correlation between two 2-D functions f and g is given by

$$p = f \odot g = \int\int f(x', y')g(x + x', y + y')dx'dy'. \quad (4)$$

It is related to the convolution integral by

$$p = f(-x, -y)^* \circledast g(x, y) \quad (5)$$

where the superscript asterisk at f means the complex conjugate of f.

Similar to the convolution, there is a correlation theorem which holds for linear transforms of f and g:

$$P = F^*G \quad (6)$$

where P,F and G are the FT of p,f, and g. Again, the asterisk means complex conjugate of F.

The correlation p measures the degree of similarity between two functions f and g. The more identical they are at a certain position (x,y) the higher their correlation value. Therefore, the correlation function is used often in template matching or even pattern recognition. An important consequence of Equation (6) is that if f or g is an even, symmetric function, the correlation is equal to the convolution.

Activity 3. Template matching using correlation.

1. Create a 256x 256 image in PAINT with the phrase "THE RAIN IN SPAIN STAYS MAINLY IN THE PLAIN"). All caps please and use sans serif fonts such as Arial or Helvetica. Let the letters be white and the background black.
2. Create a 256x256 image in PAINT of the letter "A" using the same font and font size as in step 1. Make sure "A" is in the geometric center of the image. Again, letter is white, background black. A is the "template" and we want to find matches of "A" in the target phrase image.
3. Get the FFT2 of both images.
4. Multiply the complex conjugate of FFT2(A) with the FFT2 of the phrase image.
5. Get the inverse FFT of the result in 4 and display as abs(). Comment on the result.

6. Repeat for the letter “I”. Comment on the difference between the results of using “I” and “A”.

Here’s a sample code:

MATLAB	PYTHON

AFFT= fft2(A);
P = conj(FImg).*AFFT;
#FImg is the complex fft2 of the phrase image
#Again convert the image into gray first.
smallp = ifft2(P);
figure(5);
imagesc((abs(smallp)); axis image;
	AFFT = np.fft.fft2(A) #A is the template image
	P = np.conj(FImg)*AFFT
	#FImg is complex FFT2 of the phrase image
	#Again just use one channel
	smallp = np.fft.ifft2(P)
	plt.imshow(abs(smallp))

Reference

[1] Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, *19*(90), 297-301.