Cover - 5/5
Theory - 20/20
Implementation - 10/10
Results - 50/50
Code - 5/5
Organization - 10/10

Overall - 100/100

Good job

CS 474 - Image Processing and Interpretation

Fall 2021 - Dr. George Bebis

Programming Assignment 3 Report

Written by Joseph Trierweiler and Logan Leavitt

Joseph was responsible for Experiment 1 and Half of Experiment 3

Logan was responsible for Experiment 2 and the other half of Experiment 3

Experiment 1

Theory

The Discrete Fourier Transform (DFT) is used to convert a finite series of samples from the spatial domain into the frequency domain. It takes in a sequence of real values and turns them into a linear combination of both real values and imaginary values given by complex exponential waves. These complex exponential waves can also be represented with equivalent sine and cosine waves using Euler's Equation. The equation for performing a forward DFT and Euler's equation are given below:

$$\boxed{F(u) = \sum_{x=0}^{N-1} f(x) e^{\frac{-j2\pi ux}{N}}, \quad u = 0,1,2,\ldots,N-1}$$

Equation for performing a Discrete Fourier Transform

$$\boxed{e^{\pm j\theta} = cos(\theta) \pm jsin(\theta)}$$

Euler's Equation

The Inverse DFT can be used to convert a continuous function in the frequency domain back into the spatial domain. The equation for performing an Inverse DFT is given below:

$$\boxed{f(x) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) e^{\frac{j2\pi ux}{N}}, \quad x = 0,1,2,\ldots,N-1}$$

Equation for performing an Inverse Discrete Fourier Transform

Notice how for the inverse DFT, we multiply the result by 1/N, where N is the number of real samples. This is to normalize the transform so that its results have the same number of real values equal to the amount of real samples.

In order to perform a forward DFT on a continuous function, the function must be first converted into a series of samples. These samples must be finite and equally spaced before being input into the forward DFT. The reason this is possible is because of a property of continuous functions in the frequency domain. This property is that multiplication in the frequency domain is equivalent to the string of impulses found in the continuous function in the spatial domain. These are defined in the equations below:

$$\boxed{s(x)f(x) = \sum_{k=-\infty}^{\infty} f(x)\delta(x - k\Delta x)}$$

Sampling Property

$$\boxed{s(x) = \sum_{k=-\infty}^{\infty} \delta(x - k\Delta x)}$$

Train of Impulses Function

$$\boxed{\delta(x) = \begin{cases} 1 & if\ x = 0 \\ 0 & if\ x \neq 0 \end{cases}}$$

Discrete Delta Function

$$\boxed{\delta(x) = \begin{cases} \infty & if\ x = 0 \\ 0 & if\ x \neq 0 \end{cases}}$$

Continuous Delta Function

It’s important to consider Delta X when sampling a continuous function. With more samples, Delta X will decrease and the resolution (the amount of samples in a period) of the DFT will increase. There is a computation time increase as well, so balancing accuracy with computation time is the purpose of Delta X. In order to visualize the result of a DFT, we must center the period. This can be done by shifting the resulting image by N/2, where N is the number of samples.

Implementation

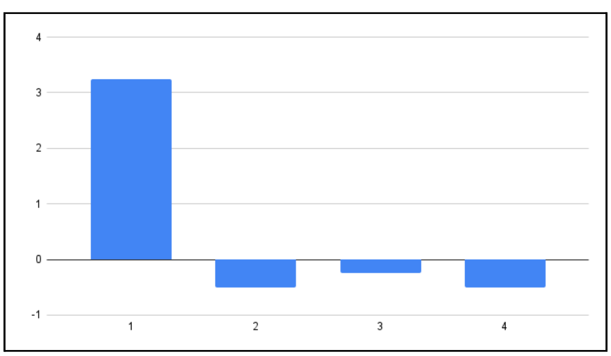
For our implementation of DFT we used an implementation of the Fast Fourier Transform (FFT). The FFT is more efficient than the DFT, with a complexity of O(NlogN) rather than O(N^2). The implementation of the FFT in our program is from a book titled “Numerical Recipes in C”. Each component of this experiment was initialized within the program, and no images were generated. In order to test the results, all that’s needed is to execute the file titled “q1.cpp”. Results from each transformation were modeled using Google Sheets, where the y axis is the value after the transformation, and the x axis is the number of samples.

Results and Discussion

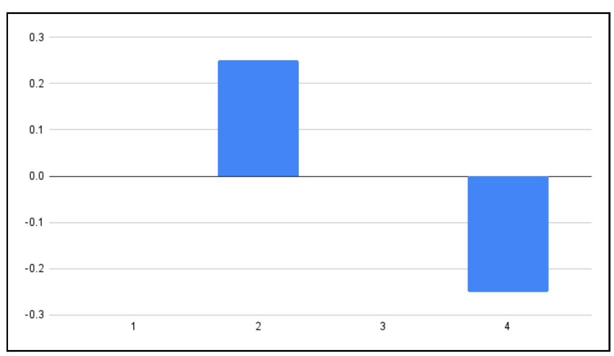
There were three components to experiments 1: A, B, and C. For component A we were given a series of samples to convert into the frequency domain. For component B, we were given a continuous function in the spatial domain to convert to the frequency domain. For component C, we were told to convert the common Fourier function for a rectangle using DFT. In each component we will show the results of performing a DFT transformation by plotting the transformation’s real and imaginary components, as well as the transformation’s magnitude and phase.

Experiment 1A - f(x) = [2, 3, 4, 4]

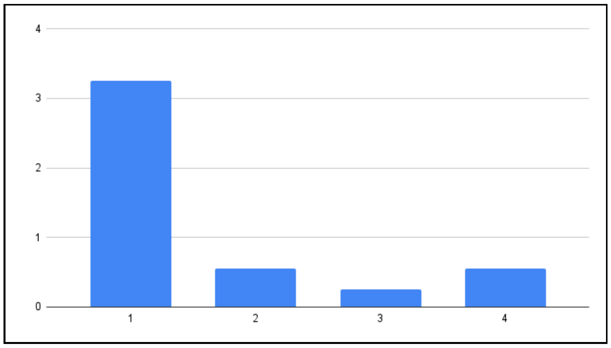
Real Component

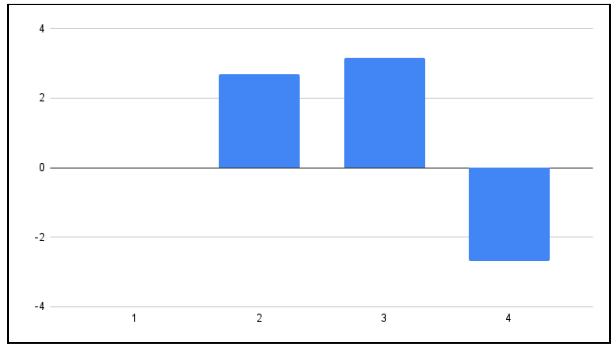


Imaginary Component



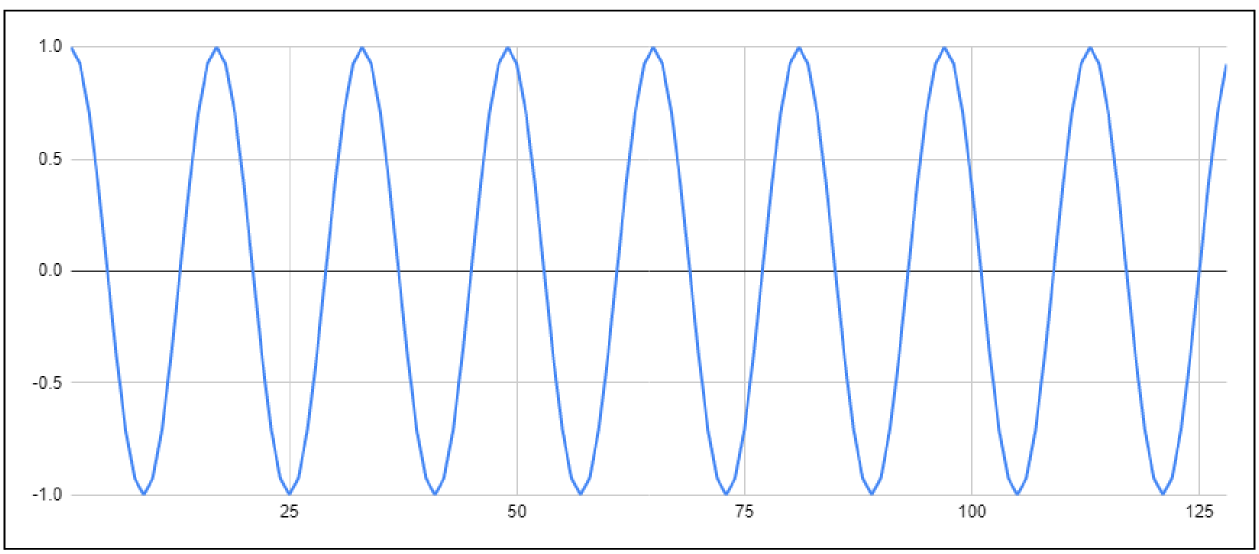
Magnitude **Phase**



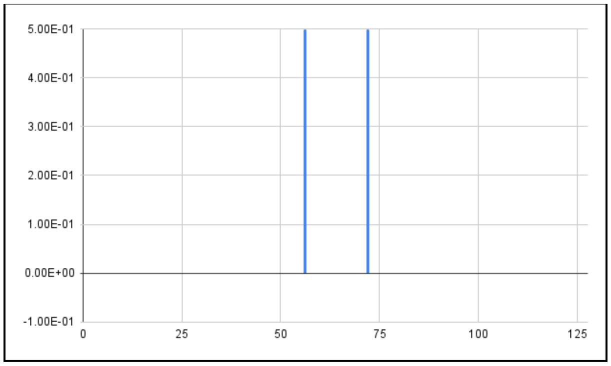


After applying a DFT on the function f(x), the results demonstrate some of the properties we've talked about in the theory section of this paper. The DFT introduces these imaginary values from the complex exponential, as shown above. It also appears that the real component of the complex exponential becomes negative at the second sample, which was not the case in the spatial domain. The magnitude of the transformation was highest in the first sample, and the phase turned negative on the 4th sample.

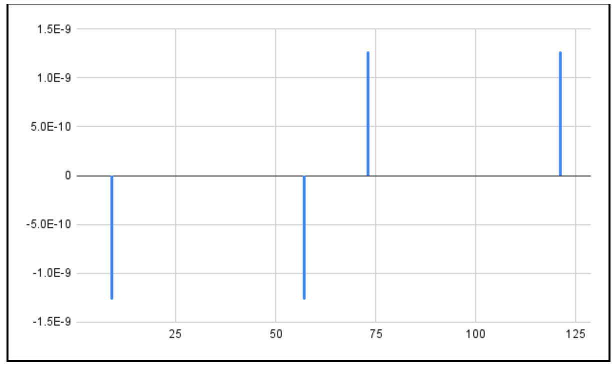
Experiment 1B - f(x) = cos(2 pi u x) where u = 8 and N = 128



Real Component

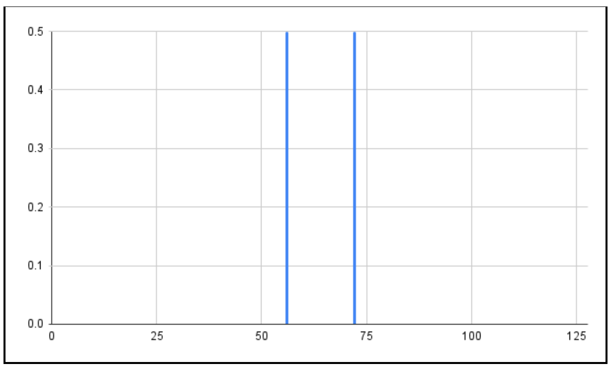


Imaginary Component

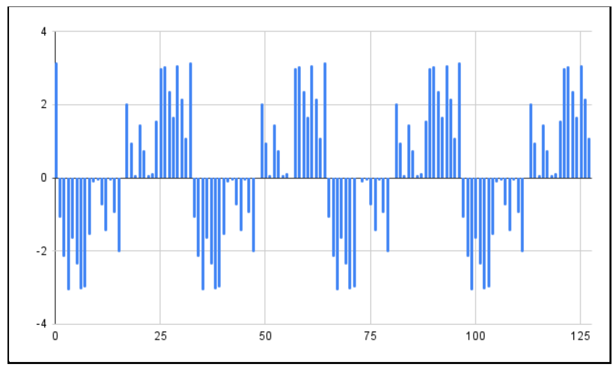


I think you should talk about the imaginary components here. Are they as expected?

Magnitude



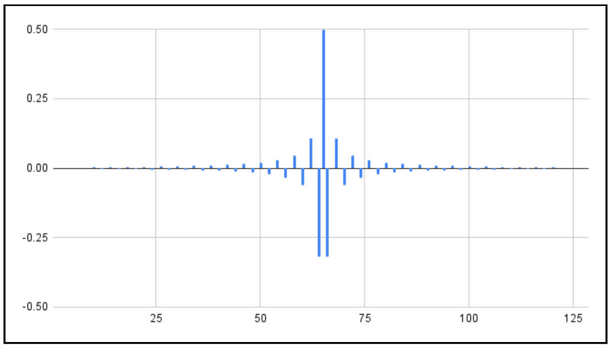
Phase



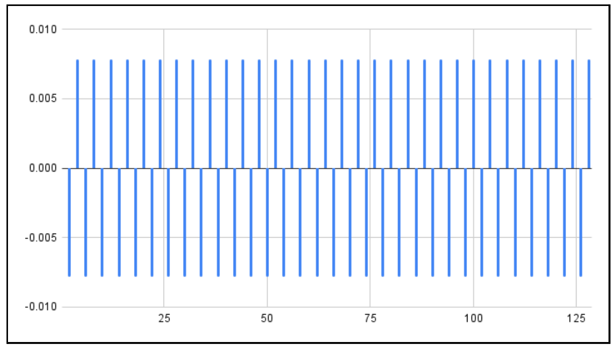
For part B of experiment 1, we were given a continuous function in the spatial domain where there are 8 cycles in one period and 128 samples. We know from our theory that the function needs to be converted into a string of impulses in order to be transformed, so with the parameters u = 8 and N = 128, we are able to perform a DFT. Similar to the previous component, we have a real component and an imaginary component. The magnitude is strongest at the 56th and 72nd samples, corresponding to where our real components are. We performed a phase shift of 64 samples (N/2 Samples) for our phase visualization. An interesting feature about the phase plot is how it follows the continuous nature of the original function.

Experiment 1C - Rectangular Function

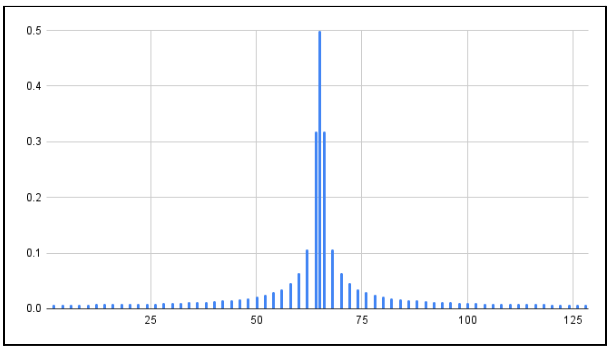
Real Component



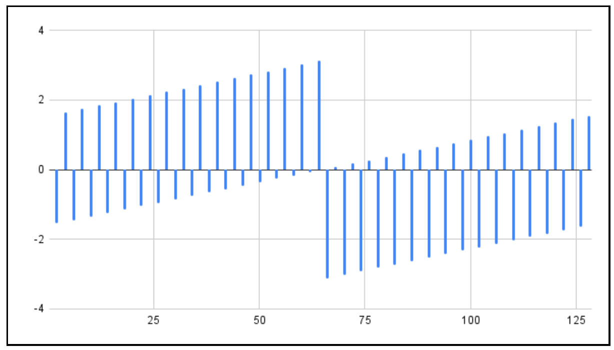
Imaginary Component



Magnitude



Phase



The DFT of the rectangle function shows one of the practical use cases of the DFT function. The real component follows the shape of a sinc function, which makes it useful since it is defined. The imaginary component inverts every sample. The magnitude continues to correlate with the real component values, except for being positive.

Should there even be imaginary values here?

Experiment 2

Theory

One key concept utilized in this experiment is the separability of the 2 dimensional Discrete Fourier Transform (DFT). For instance, consider the forward 2 dimensional DFT equation.

$$F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(\frac{ux+vy}{N})}$$

In particular, we can note that $e^{-j2\pi(ux+vy)/N} = e^{-j2\pi ux/N} e^{-j2\pi vy/N}$. This means that the forward equations can be factored.

$$F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} e^{-j2\pi(\frac{ux}{N})} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(\frac{vy}{N})}$$

The result is that the 2 dimensional DFT is just the composition of 1 dimensional DFTs. In practice, this makes it much easier to implement the 2 Dimensional DFT, assuming the 1 dimensional DFT has been implemented.

In addition to the 2 dimensional DFT, it was also necessary to shift the resulting frequencies such that the center of a period lies at the center of the image. While one method to implement this would be to manually translate values, we can instead utilize properties of the DFT. Notably, we can utilize the property that multiplying by a certain exponential term in the spatial domain results in translation in the frequency domain.

$$f(x,y) e^{j2\pi(u_0 x + v_0 y)/N} = F(u - u_0, v - v_0)$$

When $(u_0, v_0) = (N/2,\ N/2)$, the equation simplifies nicely.

$$f(x, y)(-\ 1)^{x+y} = F(u - N/2,\ v - N/2)$$

Thus, translation in this case only requires flipping the signs of certain terms, which is much more convenient and efficient to implement.

Implementation

The 2 dimensional DFT was implemented for both this experiment and experiment 3. The implementation utilizes the separability property to reduce the problem to 1 dimensional DFT's, and 'fft.c' as the implementation of the 1 dimensional DFT. The DFT is first run on each of the rows of the array, and then run on each of the columns of the resulting array. It should be noted that the routine in 'fft.c' does not multiply by a factor of 1/N, so there is no need to multiply by N after transforming the rows. Instead, in this routine, we multiply the result by 1/N after both the rows and columns have been transformed.

In addition to the 2 dimensional DFT, it was also necessary to implement the translation. This was done by iterating over each element, and multiplying by -1 if the sum of the indices was odd (i + j % 2 == 0).

Results and Discussion

For each of the square images, the DFT was applied both with shifting, and without shifting. This experiment was repeated with a 32x32, 64x64, and 128x128 square at the center of the image.

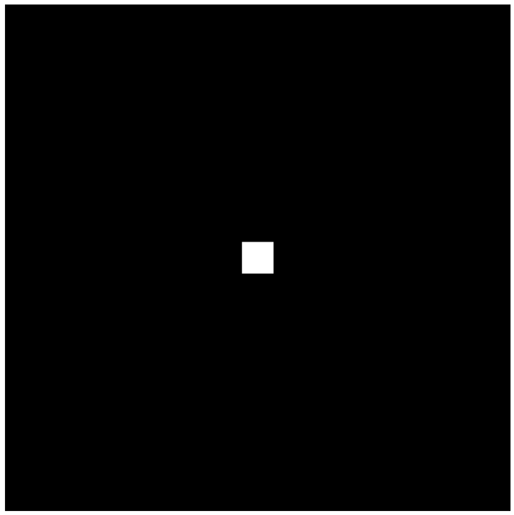


Figure 1. Image with a 32x32 square at the center.

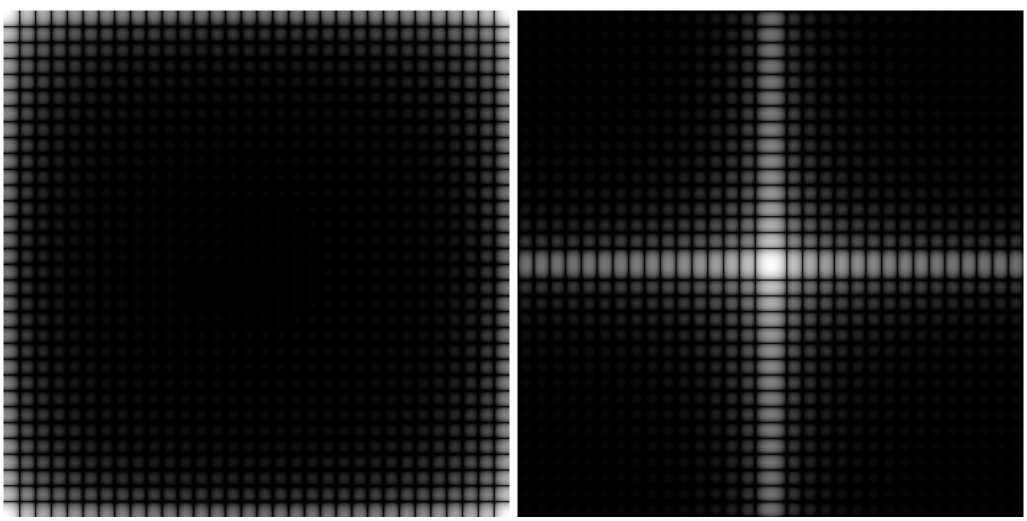


Figure 2. The resulting DFT without shifting (left) and with shifting (right). The result was visualized as $c\ log(1 + |F(u,v)|)$, where c was chosen such that the maximum magnitude is mapped to 255.

It can be noted from the results that without shifting, all of the important frequencies lie at the edges of the image, while they lie at the center of the image with shifting. This confirms the hypothesis that it is often more convenient to visualize the period when it is centered. In addition, we can also note that the DFT of the square resembles a 2 dimensional $sinc$ function, which is the expected result.

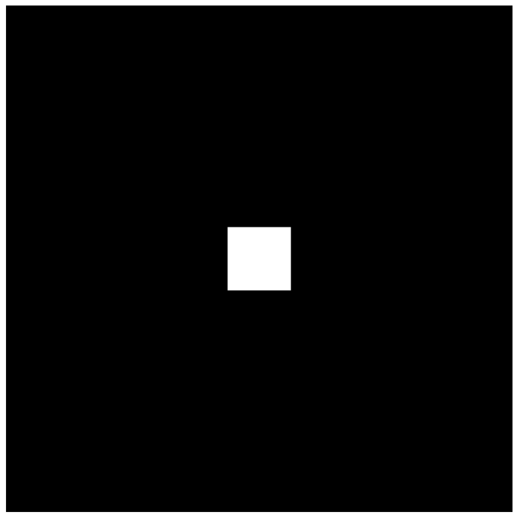


Figure 3. Image with a 64x64 square at the center.

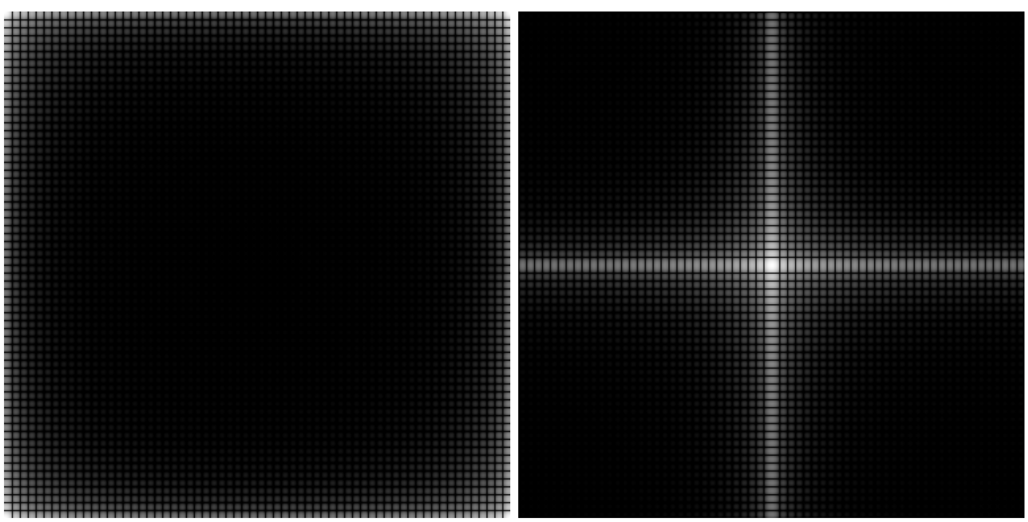


Figure 4. The resulting DFT without shifting (left) and with shifting (right).

The result was visualized as $c\ log(1\ +\ |F(u, v)|)$, where c was chosen such that the maximum magnitude is mapped to 255. We find similar results when the square is expanded. One noticeable difference is that the oscillations of the resulting $sinc$ function become more frequent as the square becomes larger, which is again confirmed when the experiment is run on a 128x128 square.

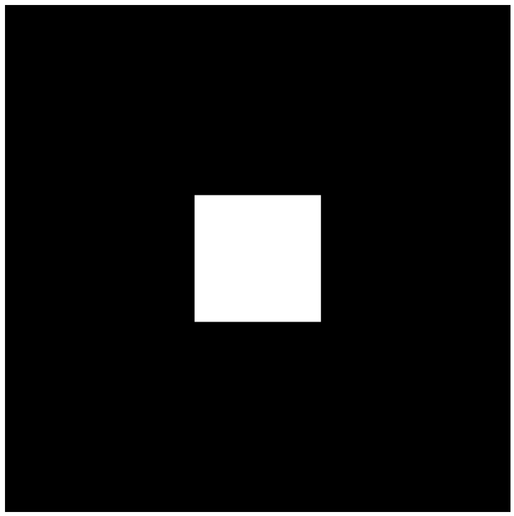


Figure 5. Image with a 32x32 square at the center.

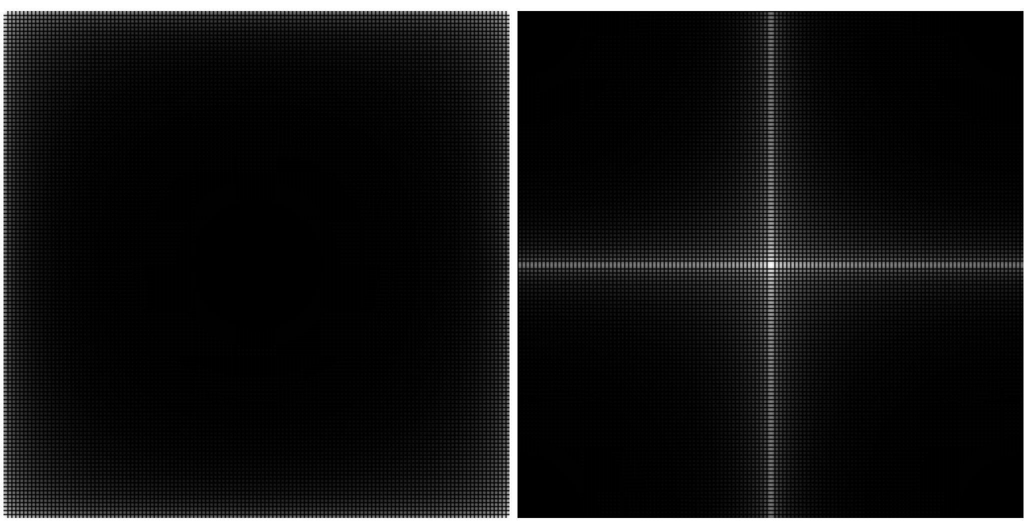


Figure 6. The resulting DFT without shifting (left) and with shifting (right). The result was visualized as $c\ log(1\ +\ |F(u, v)|)$, where c was chosen such that the maximum magnitude is mapped to 255.

Experiment 3

Theory

Each DFT can be decomposed into sinusoidal components that create its function. We can measure the phase and magnitude of these sinusoidals, which can inform us on the image's properties. Magnitude of a sinusoidal denotes the strength of that frequency in the function. Phase of a DFT describes how the sinusoidals line up. The phase of a DFT tends to relay more information about the image than the magnitude does, however, visualizing the magnitude is also useful for detecting noise and removing it.

Implementation

The implementation of Experiment 3 is similar to the implementation found in Experiment 1, except now we convert the image's magnitude and phase information into its own image. The negative values found in the phase images are set to a positive minimum value for the sake of visualization. In order to visualize the magnitude of the experiment better, we stretched the transformation by $\log(1 + |F(u,v)|)$.

Results and Discussion

Figure 7 shows the magnitude of the frequencies found in "lenna.pgm". Each pixel's quantization level represents the strength of the sinusoidals present in the image, where white is the brightest. This visualization helps us understand the properties of magnitude via being able to see its symmetric and positive nature.

Figure 7. lenna.pgm (left) and its discrete fourier transform (right) visualized as c x log(1 + |F(u,v)|) and shifted so the period is centered.

Figure 8 shows the results of performing an inverse DFT on the image, using only its magnitude. As can be seen, there is no resemblance of the image found in only its magnitude. However, if only the phase is preserved, then the edges of the original image remain. This example helps us visualize the purpose of visualizing an image's phase, and its role in image reconstruction.

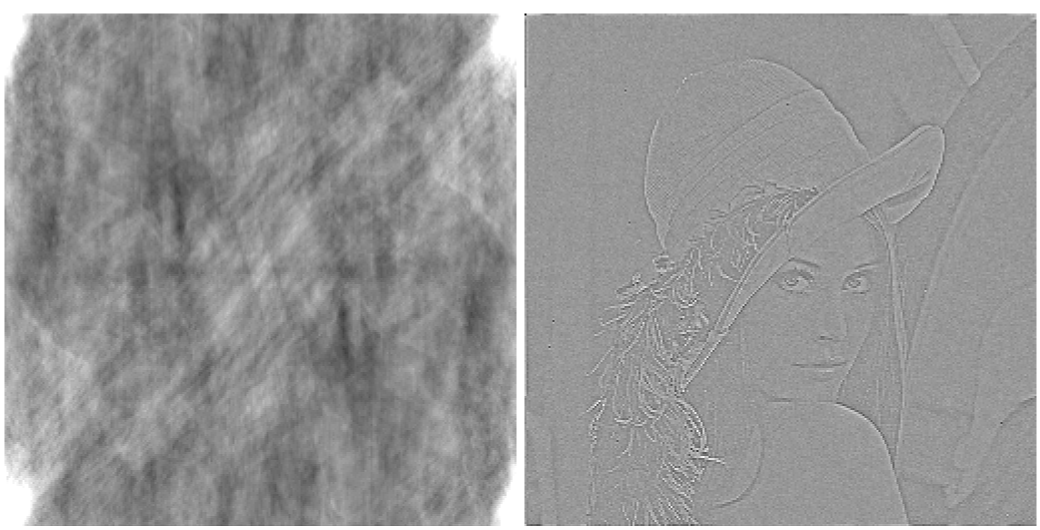


Figure 8. The inverse fourier transform of the fourier transform when (left) only magnitude is retained and (right) when only phase is retained.