Maria Shevchuk

Imaging and Image Processing

Abstract

Biomedical imaging is a rapidly growing field with a wide range of applications that involves acquiring and analyzing image data of the body on a molecular, cellular, and physiological level. Through biomedical imaging, we are able to visualize internal organs and screen them for diseases that would otherwise be impossible to detect. Some common applications include MRI scanning for various brain conditions and ultrasound imaging to examine an unborn child. In addition, biomedical imaging can be used to track the effectiveness of a certain treatment which is advantageous for choosing a patient's treatment plan.

Photodiodes can be used to measure color by converting light into electrical currents. Throughout this lab, we used an Arduino Nano Sense BLE 33 in order to acquire light signals of existing images. The data was collected with three photodiodes each detecting color intensity of a different color - red, blue, and green. When plotting the photodiode in MATLAB, the generated plot contains three lines, each one to represent one of the colors. These signals were then transformed into a representation of the real-life image sensed by the photodiodes. When working with photo images, a phone was used for data collection. Greyscale and color image data can both be represented with a matrix, however, they are 2D and 3D respectively, due to the RGB values that are associated with each pixel in a color image. Having converted our collected data into greyscale, we then used a prewritten function to obtain a 2D Fast Fouried transform of each image, which were displayed in the frequency domain. Lastly, we were able to successfully create and visualize synthetic images in MATLAB and analyze them through their Fourier transforms.

Results

<u>Using the photodiodes on the Arduino Nano Sense</u>

An Arduino Nano 33 BLE board possesses three photodiodes each detecting color intensity of a different color: red, blue, and green, by converting the energy of photons into an analog signal. The process can be more specifically described by the photoelectric effect, which takes place when a photon strikes a diode which results in an electron-hole pair. The holes will then move to the anode and electrons to the anode, in turn producing a photocurrent.

<u>Manipulating and visualizing a color signal in Matlab in 2D</u>

Time series of color data can be represented digitally in the form of a matrix of tree vectors, in which each vector serves as the color value over a set number of samples for a specific color. These vectors can be plotted over the sample number as the x axis which can be a useful visualization when comparing different color intensities (Figure 1). Images can also be digitally represented in matrix form, however, the matrix format will differ based on whether the images are grayscale or color. Grayscale images only have 2 dimensions, where the number of rows and columns is the number of pixels on the x and y axes respectively with each datapoint being the color intensity at that location. Color images, on the other hand, are 3 dimensional, with the third dimension representing the RGB (Red, Green, Blue) value at each pixel.

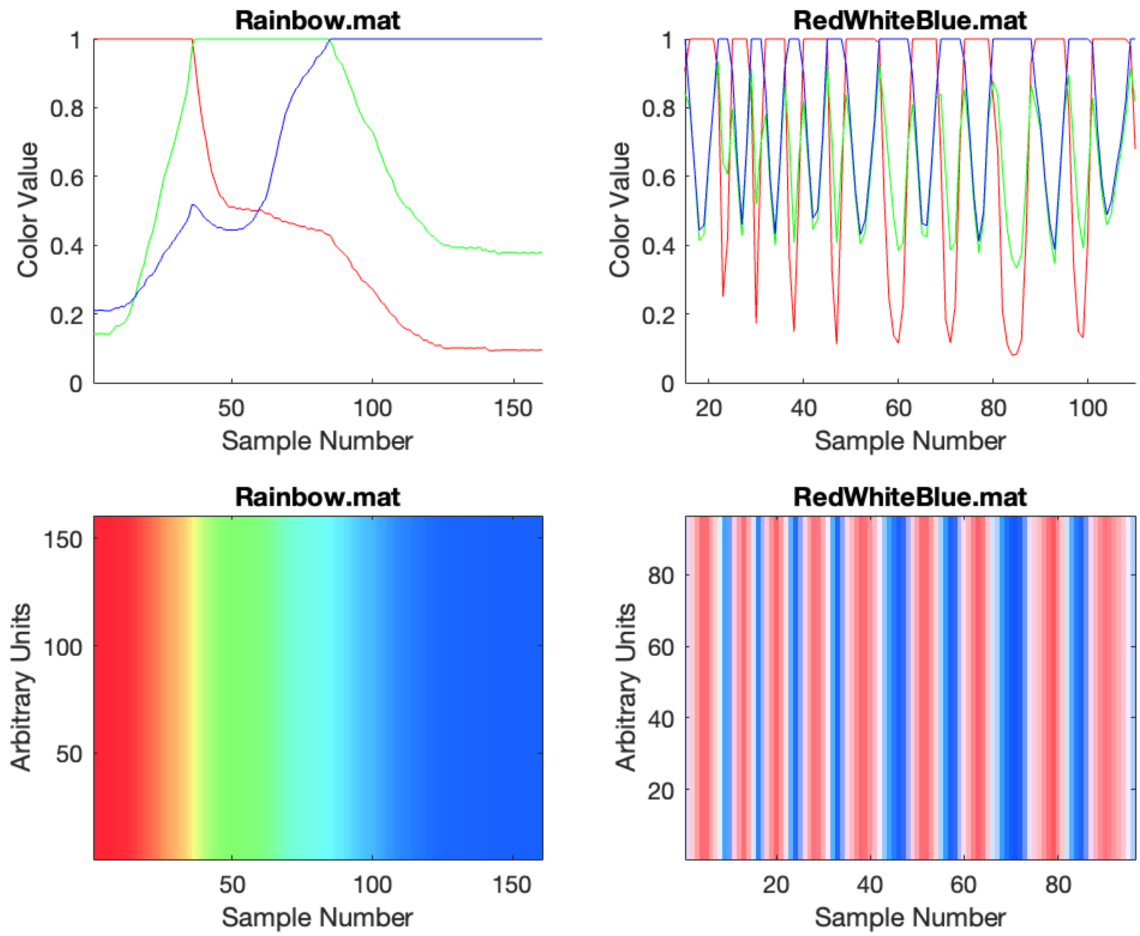

Figure 1: Acquired Color Signals and their Corresponding Transformed Synthetic Images.Top row: graph of the color value signals (red, blue, and green) each in their respective color sampled at 160 points and 95 points for the rainbow image and redwhiteblue image respectively. Bottom row: the reconstructed image using the color value signals.

Acquiring image data

In order to acquire the image data, a phone camera was used to take pictures of two already existing images (checkers and stripes). First, the pictures were taken at the original orientation and then the camera was rotated at a 45 degree angle to obtain the second striped image, and then zoomed in for the second checkered image. These photos were then saved as png files and uploaded to MATLAB. Then, each image was normalized. This process changes the range of pixel intensity values in order to equally scale the data. Mathematically, normalization is achieved by subtracting the mean pixel value and dividing the result by the standard deviation of the pixel values. All images were then displayed using the imagesc function. The checkerboard image obtained is the 2D analog of a 1D square wave signal, where the black boxes have really low pixel intensity, while the white boxes - high intensity, equaling the amplitude of the square wave.

<u>Examine image data in the frequency domain</u>

By looking at the 2D Fourier transform we are able to determine the geometric characteristics of a spacial domain image, even without access to the image itself. The Fourier transform of the checkerboard image is symmetric about x=y which can be seen in Figure 2. This makes sense since checkerboard images are symmetric and have repeating patterns in botht the x and y axis. It is also evident that there is a greater concentration of pixels with lower frequencies because they repeat more frequently throughout the image than those with higher frequencies. The big (zoomed in) checkerboard image is very similar, yet due to the fact that there are fewer repeating patterns throughout the image, there are lower densities of the lower spatial frequencies. The 2D Fourier transform of both of the checkerboard images is equivalent to the 1D Fourier transform of a square signal wave. Moving on to the vertically striped image, the frequency domain displays evenly spaced out streaks along the X axis. This can be explained by the fact that in solid vertical lines, the Y pixel frequencies make up all frequencies and any pixel frequency will make up a vertical line. Similarly, the diagonal lines are made up of both x and y pixels at various frequencies, which is why both vertical and horizontal streaks are observed in the frequency domain of the image.

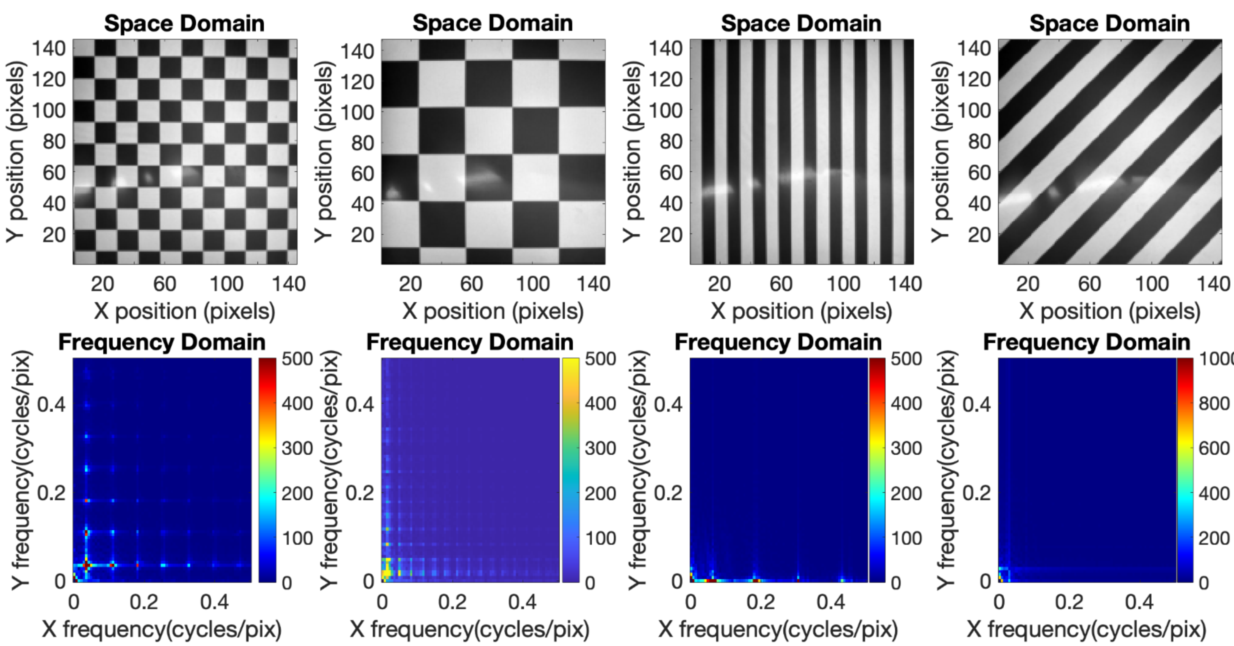

Figure 2: Images and their corresponding frequency domains. Top row: space domain graphs of small checkerboard, big checkerboard, vertical lines, and diagonal lines images respectively. Bottom row: spatial frequency domains of the images. The frequency domain was calculated by using a 2D Fourier transform function in MATLAB.

Creating and visualizing synthetic images

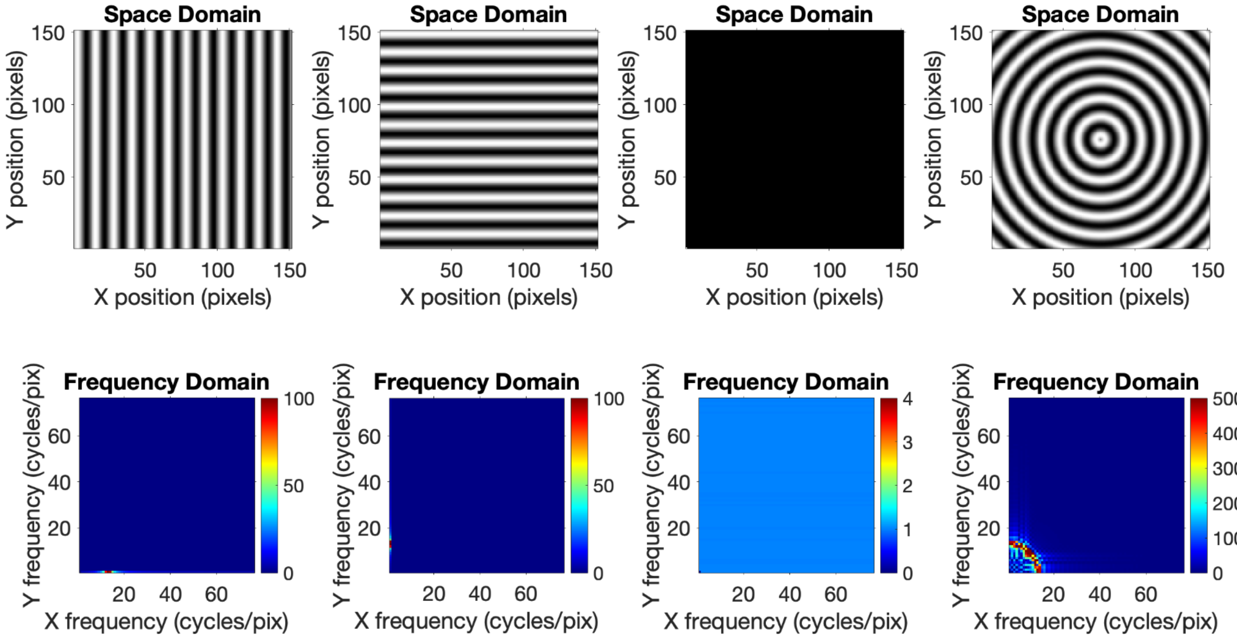

Figure 3: Synthetic images and their corresponding frequency domains. Top row: space domain graphs of four synthetic images generated in MATLAB. Bottom row: spatial frequency domains of the images. The frequency domain was calculated by using a 2D Fourier transform function in MATLAB.

In the final part of the lab, we worked on creating and visualizing synthetic images. More specifically, we generated four synthetic images and then analyzed their Fourier transforms (Figure 3). The Fourier transform of the first generated image (vertical stripes) displays a range of intensities along the x axis, but not beyond that. This demonstrates that there is a repeating frequency of pixels along the x axis of the synthetic image, which is true since the lines are vertical. As for the horizontal stripes, the Fourier transform is a rotated verion of that for the vertical stripes, with intensities shown across the y axis instead. This is logical because the second image is simply a 90 degree rotation of the vertical stripes image. The third generated image is completely black with the exception of a single bright pixel in the bottom left corner. The Fourier transform of this image is equivalent to a 1D impulse wave due to the equal magnitude across all frequencies. The last synthetic image contained concentric circles, in which case the Fourier transform was shaped as a circle at the origin of the plot, with sinusoidal peaks ranging from 0-0.1 cycles/pixel.

Discussion

Biomedical imaging is essential in many fields of medicine. It provides a way to diagnose patients and determine whether a treatment is effective or requires modification. Common medical applications include MRI, ultrasound, CT scans, etc. Biomedical imaging can also be a way to minimize human-error and potentially prevent misdiagnosis using new machine learning classification methods. When used as a tool for research, biomedical imaging can be used with images that range in scale from molecular images to human imaging and even larger. In this lab, we were successfully able to visualize and manipulate image data with little trouble and no discrepancies from our expectations. We did run into some technical issues when working with MATLAB, specifically choosing bounds for the color axis, but they were promptly addressed and resolved.