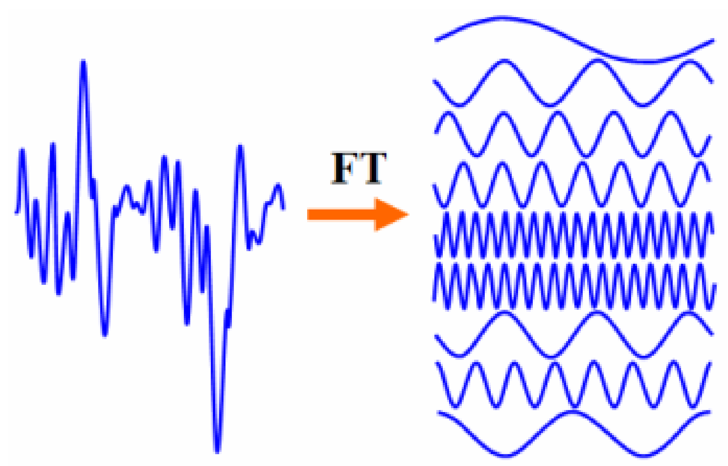
ImageCloud: An Investigation of Fourier Transform through Image and Sound

Patrick Huston, Kai Levy, & Keenan Zucker || Olin College Spring 2015

Introduction

The Fourier series is a mathematical process in which periodic functions are written as sums of sines and cosines, as seen below. This can be applied to a Fourier transform, by which this process approaches infinity. This theory and application has been used across many domains, two of which being image compression and sound processing. For our final project, we decided to apply both of these functions. Our goal was to convert an image into sound, upload SoundCloud, and be able to revert the sound back into an image.



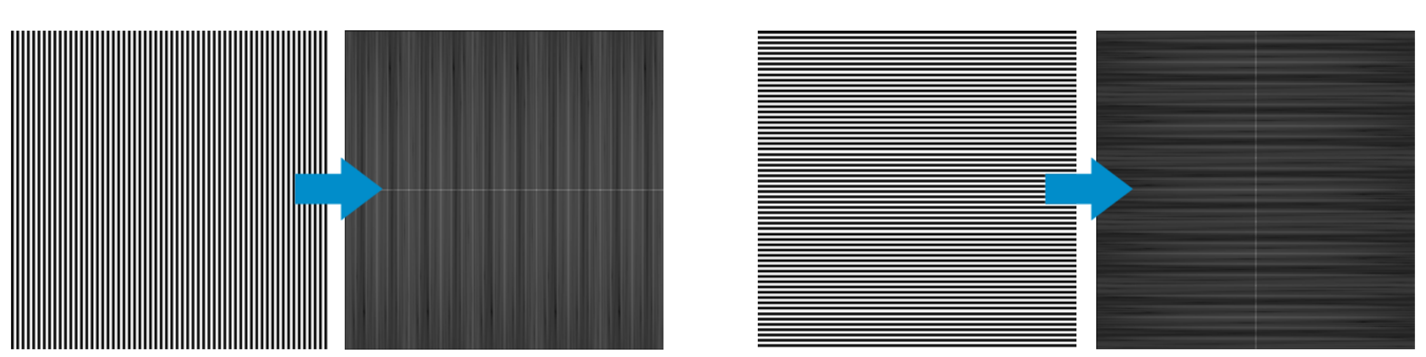
to

The Process

Our first step was to convert the input image into the frequency domain. This meant that we had to turn the image into an array. The brightness of every pixel in the image can be thought as representing discrete values in a function that 'describes' the image. To obtain a frequency domain representation, a Discrete Fourier Transformation (DFT) is taken along the two dimensions of the image, which results in a two-dimensional view of the frequency domain. The equation is as follows:

$$F(k,l) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j)\, e^{-\iota 2\pi \left(\frac{ki}{N} + \frac{lj}{N}\right)}$$

To illustrate this process, two images of horizontal and vertical stripes are used as inputs. In each of these images, one of the components (vertical or horizontal) can be approximated fully by a flat line. The other component of the image is a square wave, and the stripes in the frequency domain represent the different frequencies of sine waves.



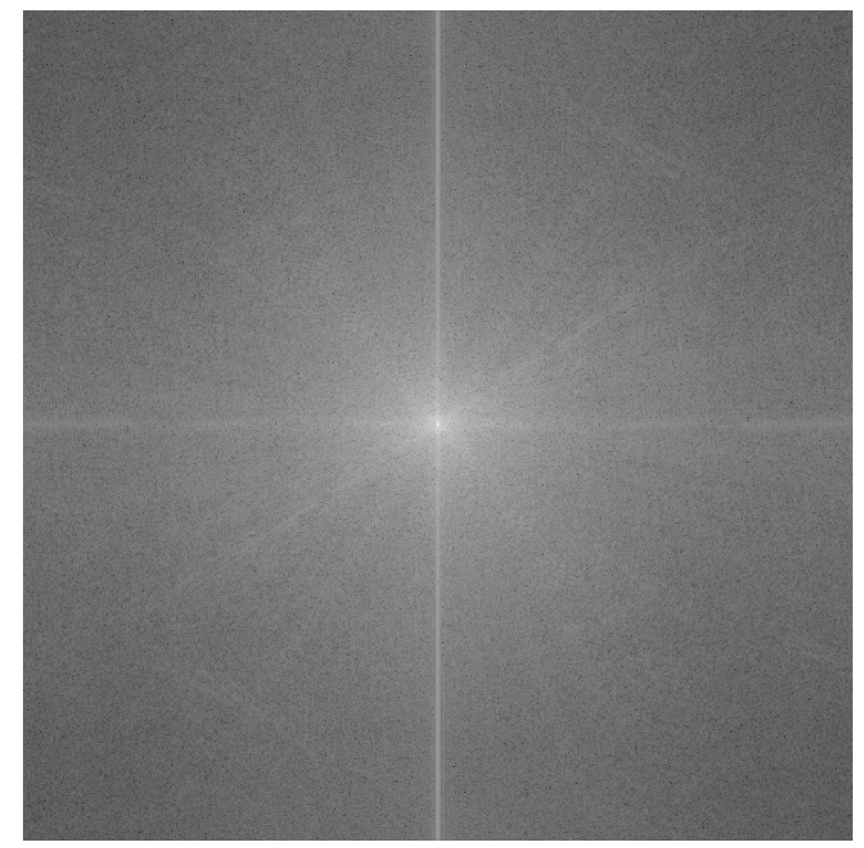
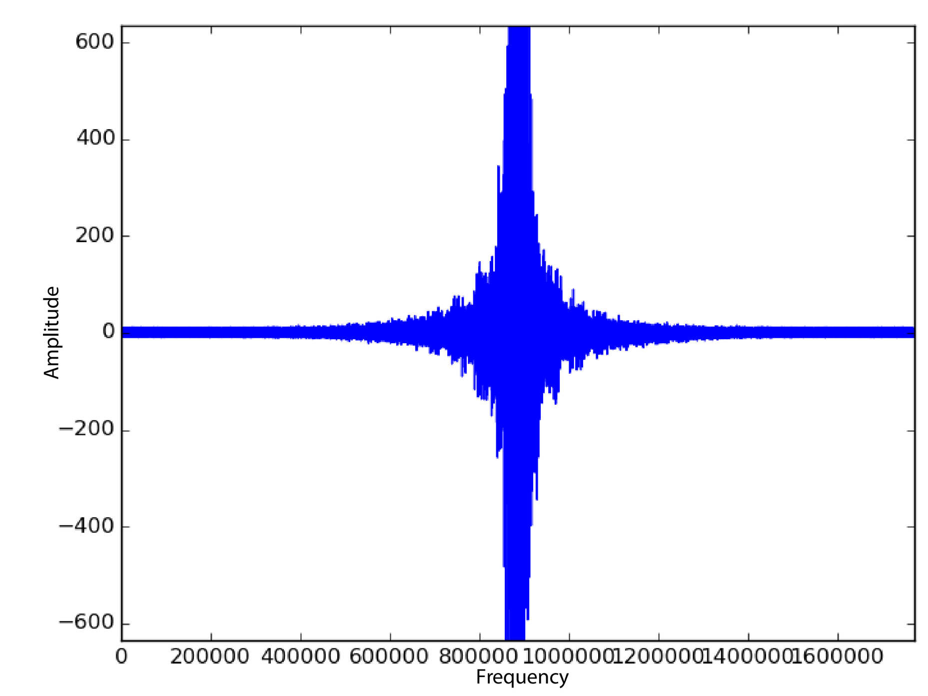
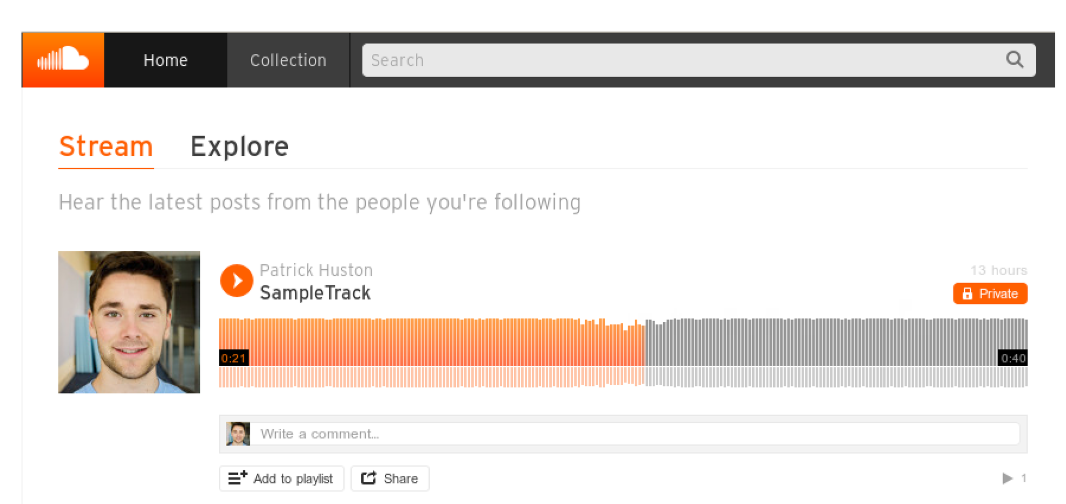


Image is converted to magnitude spectrum using DFT, shown as frequency domain.

Inverse DFT is taken. 2D array is flattened into 1D. Goes sequentially using rows of frequency domain.



Audio file is generated from 1D array of frequency domain and uploaded to SoundCloud.



WAV file can be downloaded from SoundCloud and converted back into an image.

How We Did It

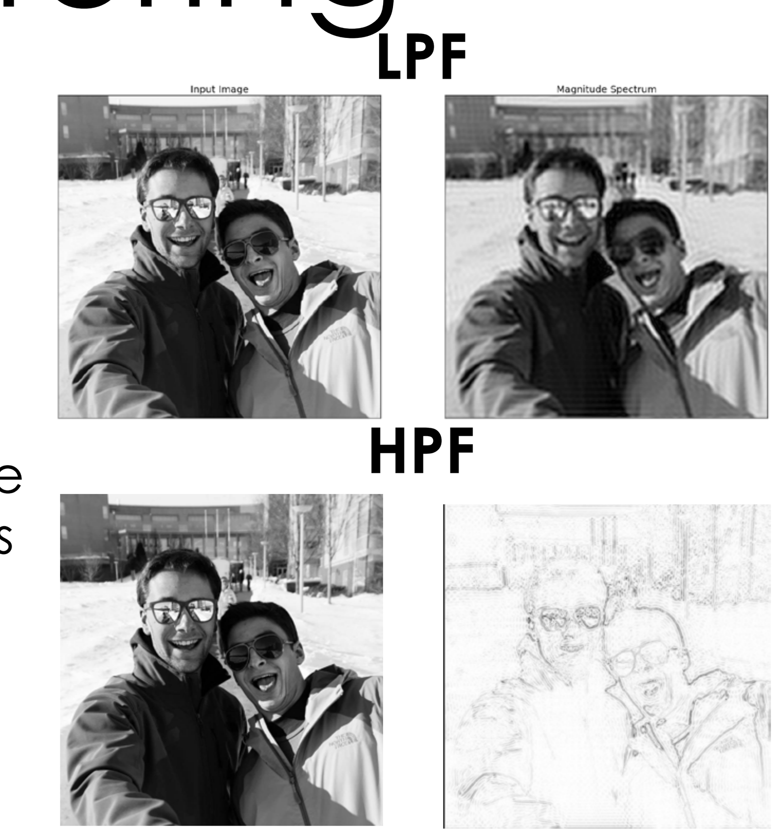
We used the powerful Python language for this project. We used several libraries to assist our image processing and sound conversion. These were: NumPy, OpenCV, Matplotlib, and SciPy.

Our Findings

Images don't produce nice melodies! The sounds we generated were loud and harsh. It was a very interesting process to get there, but it wasn't music to our ears.

Filtering

We also implemented filtering into our program. The top image shows a Low Pass Filter applied on the image, blurring the contours. The bottom image shows the High Pass Filter, which exaggerates the contours of the image.



Future Work

For future iterations, we would like to work with color. We think we could implement different sounds based on color, creating more unique noises, as well as turning full songs into images

References

[1] Lehar, Steven. "An Intuitive Explanation of Fourier Theory." Boston University, n.d. Web. 03 May 2015.

[2]Mordivintsev, Alexander, and Abid K. "Fourier Transform¶." Fourier Transform — OpenCV-Python Tutorials 1 Documentation. N.p., 31 Oct. 2014. Web. 03 May 2015.