Project Freedom : Merging Audio & Images With FFT
Executive Summary

Gabriel Aldous
MAS4115
Project on GitHub

Project Overview

Project Freedom is a easy to use CLI program that users can interact with in order to perform their own experiments with the Fast Fourier Transform and related operations on both images and audio files. Basic filtering operations are supported for images and audio, and there are numerous ways to visualize how the files have changed and preview those changes. Additionally, the application supports easy modification of the file system, and the classes employed to perform these image modifications can easily be repurposed by other programmers to add additional functionalities.

In addition to the basic manipulations, there is an implemented advanced 'hybridization' technique for both audio and image files. Image files can be 'hybridized' so that the image appears to be one thing up close, and another when viewed from far away, while audio files can be modified to make it appear that one sound is being heard within a different environment.

This project is implemented in Python, and supports building on Python versions 3.9-3.12 (though it was only tested on version 3.12). It relies on Numpy, Matplotlib, Scipy, and PyGame, and instructions for setting up and running the program are available in the README file. To view the results of various operations performed by Project Freedom, see the github repository.

Mathematical Principles

The fundamental operation which this program seeks to foster a greater understanding of is called the Fourier Transform, or more specifically the Fast Fourier Transform (FFT). This technique is often used in signal processing applications, as is the case in this project. The FFT is utilized to turn some signal data (such as an image or an audio file) into a different domain called the frequency domain. Simply speaking, the FFT takes the input and restructures it so that information about which frequencies are more prevalent is easily accessible.

In the context of the custom hybridizations, audio hybridization is performed using a convolution, while image hybridization is performed by filtering the high detail portions of one image (the edges & lines) and overlaying it on the low detail portion of another image (the general colors & shapes). This results in an image which can trick human perception and appear differently from different viewing distances.

The FFT has applications in many areas, notably in image compression formats like JPEG and MP3, filtering operations, wireless communication, and much more. For more details about how the FFT works, and the linear algebra supporting it, view the .