Report for the Sonification Tool FotoAudify

Aman Prajapati

29 December 2024

Contents

1 Introduction

Sonification is the process of converting data into non-speech audio signals. The goal of this tool (named as FotoAudify) is to enable users to transform astronomical images into audible sound. Using image pixel data, specifically RGB values, the tool generates unique audio based on brightness and color mappings.

This report details the development process, functionality, and implementation of the sonification tool based upon the requirements of the problem statement.

2 Objectives

The main objectives of this project are:

- To extract pixel-level RGB data from astronomical images.
- To provide multiple sonification modes (e.g., brightness-based pitch modulation and color-based sound effects).
- Create a simple user interface for easy interaction.
- To enable users to save the generated sonified audio files locally in their system.

3 System Architecture

The tool consists of the following key components:

1. **Image Processing Module:** Extracts RGB pixel data from uploaded images.
2. **Sonification Module:** Maps pixel data to audio frequencies.
3. **Command line Interface:** Allows users to upload images, choose the sonification modes and save the generated audio files.

4 Implementation

4.1 Image Processing

Images are resized to a fixed resolution (specifically 200x200 pixels) for uniform processing. The data of the pixels are then extracted and stored in a CSV file with the format: `Row, Column, Red, Green, Blue`.

4.2 Sonification Modes

1. **Brightness-Based Pitch Modulation:** Maps pixel brightness to pitch frequency.

2. **Color-Based Sound Effects:** Maps RGB channels to different audio characteristics.

4.3 Web Interface

A Flask-based web interface will be developed in near future to allow users to access this tool via a website. Users will be required to:

- Image Upload

- Mode Selection

- Audio File Download

5 Code Example

Below is an example snippet for image processing:

Listing 1: Image to CSV Conversion

```
import cv2
import csv

def image_to_csv(image_path, csv_path):
    image = cv2.imread(image_path)
    image = cv2.resize(image, (200, 200))
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

    height, width, _ = image.shape

    with open(csv_path, ’w’, newline=’’) as file:
```

```
writer = csv.writer(file)
writer.writerow(['Row', 'Column', 'Red', 'Green', 'Blue'])

for row in range(height):
    for col in range(width):
        r, g, b = image[row, col]
        writer.writerow([row, col, r, g, b])
```

6 Results

The tool successfully:

- Processed astronomical images into pixel-level CSV data.
- Generated distinguishable audio outputs based on brightness and color mappings.
- Generated and saved the audio file .

7 Conclusion

The astronomical image sonification tool bridges the gap between visual and auditory data representation. Future enhancements could include additional sonification modes, improved audio mapping techniques, and support for larger image datasets.

8 Future Goals

- More efforts can be made towards cleaning the audio file from the noise present in the generated audio files to create a melodious sound.
- Code can be optimized to process high resolution images
- A web interface can be developed so as to enable users to try sonifying astronomical images captured through their cameras

9 References

1. Python Documentation: `https://docs.python.org/3/`
2. Some Youtube videos