Spring 2020

Pixel Composition: Converting Images to Music

Eric Anderson
Western Washington University

Follow this and additional works at: https://cedar.wwu.edu/wwu_honors

Part of the Music Commons

Recommended Citation
https://cedar.wwu.edu/wwu_honors/391

This Project is brought to you for free and open access by the WWU Graduate and Undergraduate Scholarship at Western CEDAR. It has been accepted for inclusion in WWU Honors Program Senior Projects by an authorized administrator of Western CEDAR. For more information, please contact westerncedar@wwu.edu.
Pixel Composition: Converting Images to Music

Investigating how to generate music from color data in images

Eric Anderson

Advisor: Dr. Charles Halka
Algorithmic Composition

- Creating music using algorithms or procedures
  - In the Information Age: determined by mathematics or logic
  - Computer is given instructions on what to produce

- Two Methods:
  - Continuous Output: waveforms (like ambient sound)
  - Discrete Output: notes on a music score
    - This is what I’m investigating
    - **Objective**: create legible music that could be performed by musicians

- Need to define “Music” in the simplest terms
  - Collection of discrete frequencies organized to fit a pulse
  - Western Music Theory, not exclusive
Other Research

- Image Sonification
  - Audio communicates visual message or portrays data audibly
  - Spectrograms (viewing music as an image) are standard

- Markov Chains
  - Assigning musical elements probabilities of occurring after another

- Machine Learning (non-AI)
  - Training a computer based on previous human materials
  - I believe this limits potential
My Approach

- Create my own computer program that can analyze an image, and generate music from its pixel data.

- Images: hyperspectral data
  - R/G/B format, 24-bit (0-255)

- Music: MIDI data
  - Musical Instrument Digital Interface
  - 128 values
Tools

- Coded in R (open-source statistics programming language)
  - “imager” package
    - This made it super simple to store images as numerical data and transfer it to a MIDI file for playback

- MIDI-to-CSV and CSV-to-MIDI
  - John Walker, Fourmilab
  - Translates MIDI file to a CSV file, and vice versa

- ImageMagick
  - Image Editor, native R functionality
    - Allowed me to create a video showing what pixel was being played by the music
Here’s My Program!

- My true final product

- Made available publicly on GitHub
  - My instructions and code comments (in green) will help other users
    - This is computer science etiquette
Procedure

- Impractical to analyze every pixel
  - Take equidistant samples
  - User chooses how many rows + columns of pixels to extract

- **Example:**
  - 50 rows, 50 columns
    - 2,500 pixels
  - R/G/B channels converted to three sets of MIDI notes
    - Random note range defined by the user
Procedure (cont.)

- Choose a path!
  - Four possible pathways
  - In this example, we take a spiral outwards pattern
Procedure (cont.)

- Build the CSV file
  - Program terminates when this file is populated
  - Metronome is added to aid the listener
  - CSV-to-MIDI converts the CSV into a MIDI file for playback

```plaintext
[1] "Radial pattern from center to outside. # of notes: 2500 . # of movements: 19 ."
[1] "MIDI note range: 28 - 92"
[1] "Quarter BPM: 116 ; Time Signature: 2 / 4"
[1] "Key Signature: G Minor"
[1] "Instrument 1 = Piano" "Instrument 2 = Piano" "Instrument 3 = Piano"
[1] "Tempo changes: OFF"
[1] "Grayscale volume: ON"
[1] "Est. duration of piece: 25.1 minutes"
Time difference of 3.9 secs
```

2 minute sample audio
Back to an Image!

And Video!
Creating a Score

- Limited to the quirks of MIDI – how a notation software has to make guesses, even with quantization settings on
  - Lack of articulation / other musical embellishments. This is a whole different beast.
Another Example:

Pixel Composition: Converting Images to Music

- 2264 x 3017 pixels
- Sampled Pixels
- 112 x 150 pixels

- 2.16 hours of music (traversing rows)
- Runtime: 1.2 minutes (26 minutes extra to create the video)
Another Example:

Runtime: 4.4 seconds (14 seconds extra to create the video)
Can we improve the quality of the music and images?
- Music = tonality
  - Tendency to be central & stable around one pitch (the tonic)
- Images = beauty
  - How we process the color palette

Simplify the media
- Cannot rely on the computer to make it more complex
- Music: shift notes to a diatonic scale (7 notes instead of 12)
- Images: reduce color palette to 6-bit (4 options per color)
- Hopefully, simpler music and color will make the outputs better fit our expectations
Improvement Attempt

- Key Signature: A minor
- Spiral outward

50 x 50 pixels
Tonal Music
6-bit Color
Multiple Conversions

- Multiple iterations lead to extreme degradation
  - Sloppier and sloppier until only black and white
    - This means only two notes – extreme high and low

Original  One Iteration  Five Iterations
Music to Image

- **Brahms**: Trio for Clarinet, Cello, and Piano
  - 19 minutes long, 4,590 notes
  - Spiraling inward path

- Looks neat, but nevertheless random

- Can this be improved with multiple conversions?
  - Storage differences

51 x 90 pixels
(random factor combination of 4,590)
Includes making and saving the image showing which pixels were extracted. Excludes adding metronome.

5 simulations each.

Rectangular photos, spiraling patterns take 10%-15% longer.

Most optimal/intuitive number of pixels to extract: 10,000 (100 x 100).

Tradeoffs? More music or more image?

---

### Runtime of Image.R, by Image Size

<table>
<thead>
<tr>
<th>Image Size (approx.)</th>
<th>Runtime (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>3.9</td>
</tr>
<tr>
<td>Medium</td>
<td>4.8</td>
</tr>
<tr>
<td>Large</td>
<td>8.2</td>
</tr>
</tbody>
</table>

### Duration Music Generated, by Image Size

<table>
<thead>
<tr>
<th>Image Size (approx.)</th>
<th>Duration of Music (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>518.0</td>
</tr>
<tr>
<td>Medium</td>
<td>553.3</td>
</tr>
<tr>
<td>Large</td>
<td>564.9</td>
</tr>
</tbody>
</table>
Further Exploration

- Potential Additions:
  - More advanced image processing techniques
    - Edge detection (find dramatic differences in neighbor pixels)
  - Reading vector images: points, lines, polygons
  - Improve final video visualization
    - Flash the colors being played at that moment
  - Better definition for rhythm
    - The community may have an answer
Art vs. Logic

- Embedded in my college education

- Was my project an art piece?

- Pixels have a job. Music notes have a job.
  - Flawless conversion between mediums impossible

- Imagine if I challenged you to do what I expected of the computer

- Computers will output so much garbage, but we might find a diamond in the rough
Questions? Comments?