SSIMAGESS: SIMULATING SYNESTHETIC IMAGE-TO-MUSIC AUDIO GENERATION FOR EXCELLENTLY STIMULATING SOUNDS

Lior Arad, Tanay Mannikar, Blake Schwartz

University of Texas at Austin

ABSTRACT

In this project, we propose Simulating Synesthetic Image-to-Music Audio Generation for Excellently Simulating Sounds (SSIMAGESS), a system to convert textures, lighting, shapes, and colors from images into auditory representations as musical phrases. Many synesthetes have links between visuals and audio, often experiencing both at the invocation of one. By translating spatial textures, orientations, and colors into interpretable and well-justified auditory representations, we are able to model specific synesthetic relations to a degree of accuracy with great interpretability. Our program successfully uses image features to produce pseudo-generative minimalist compositions that reflect consistent perceptual attributes of the input image.

1. INTRODUCTION

The objective of SSIMAGESS is to convert objects present in an image into evolving musical voices, or components, based on the connections found in a specific form of synesthesia. This poses an interesting challenge, since music is a dynamic temporal phenomenon while images are static representations of spatial information. SSIMAGESS has multiple phases that an image is passed through to generate synesthetically pleasing music. First, the input image is segmented into unique objects and a depth map is calculated to gain distance information for each segment. Second, most prominent colors are extracted as musical keys from each segment and visual texture properties are calculated which map to features in the timbre space. Finally, the intermediate parameters are passed to a minimalist music generation system.

2. AUDIOVISUAL MAPPING

Perceptual attributes and technical measures of images are mapped to musical characteristics to be assigned in our audio synthesis program. The basis for these mappings derives from how visuals are experienced by an individual with synesthesia like Tanay.

2.1. Synesthesia Background

Synesthesia is a neurological phenomenon found in certain individuals that have inherent connections between two or more unrelated sensory modalities such as sound with vision (chromesthesia), letters and numbers with color (grapheme-color), or the rarer word to taste (lexical-gustatory). Usually the connections are fixed, meaning they are not subjective by individual and are not inherited, although the ability to possess a given form of the condition itself is usually inherited [1]. In the case of chromesthesia, sounds are perceived to have inherent phantom visual attributes such as color,

brightness, texture, depth, movement, and other visual attributes. These parallels may differ based on the pitched or unpitched nature of a sound, sound intensity (loudness), and timbral qualities among other factors.

2.2. Audiovisual Parallels

In our project, musical key, or harmonicity for sake of generality, maps to color, while timbral, percussive, or inharmonic qualities map to visual texture. Other attributes such as dynamics, compression ratio, panning, and octave class correspond to perceived depth and location in the perceived visual space. These along with other qualities not assessed in scope of this project define Tanay's primitive audiovisual mapping, as he has chromesthesia and possesses well-defined connections between the auditory and visual domains.

2.3. Circle of Fifths

The circle of fifths organizes the 12 pitches in the standard Western 12-tone equal temperament tuning system into a circular orientation where consecutive pitches are separated by fifths. The outer ring usually organizes major keys by root note such that a full rotation around the circle corresponds to a complete modulation cycle of major triads with root note incremented by fifths, returning back to the original triad after 12 modulations. This construction is repeated in the inner circle, which realizes the same process for minor keys. Each major key is located closest to its relative minor such that the two concentric circles correspond to the relative distance between two triads that define a given pair of keys.

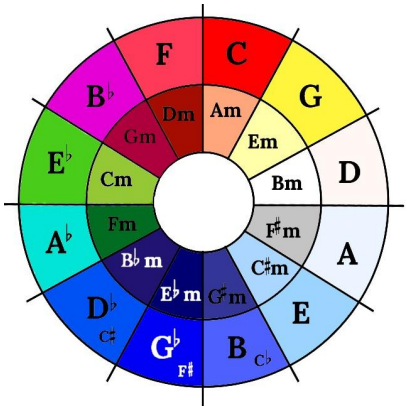


Figure 1: Tanay's Chromesthesia Circle of Fifths

Figure 1 visualizes Tanay's unique circle of fifths which denotes the colors he associates with each key or triad. One notices that many of the relative major and minor keys have a similar hue with a lower saturation for the minor key in each pair, and that local shifts in hue are mostly minimal in the hue space on the additive color wheel in Figure 2 with some apparent discrepancies [2]. The importance of hue and saturation in determining a color's closeness to a certain key is addressed in Section 4 when developing a perceptual color distance metric.

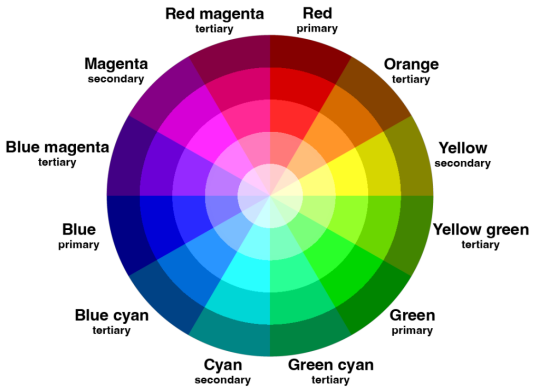


Figure 2: Additive Color Wheel from [2]

2.4. Perceptual Timbre Space

Musical timbre is defined broadly as the characteristic of a sound independent of pitch or intensity. Timbre is usually associated with the spectral qualities of sound. These include relations between overtones and partials, the ADSR envelope (attack, decay, sustain, release), and previous exposure to other timbres or masking by overlapping sounds [3]. However, most timbres can be generally defined by three perceptual auditory attributes: spectral centroid, spectral flux, and log attack time [4]. Spectral centroid is calculated as the centroid frequency of the frequency spectrum of a given timbre, disregarding the temporal changes present. Spectral flux is defined as the change in successive frequency spectra of a sound, usually visualized as a spectrogram. This is computed by taking the Euclidean distance between successive windowed DFT vectors of the sound and can be normalized depending on the application, encoding the time-varying aspect of timbre. Log attack time takes the logarithm of the attack characteristic of the signal envelope, or the time from onset to maximum intensity. These three properties are used to define a three-dimensional *timbre space* as shown in Figure 3 to map these time-frequency attributes to instruments and synthesized sounds [5].

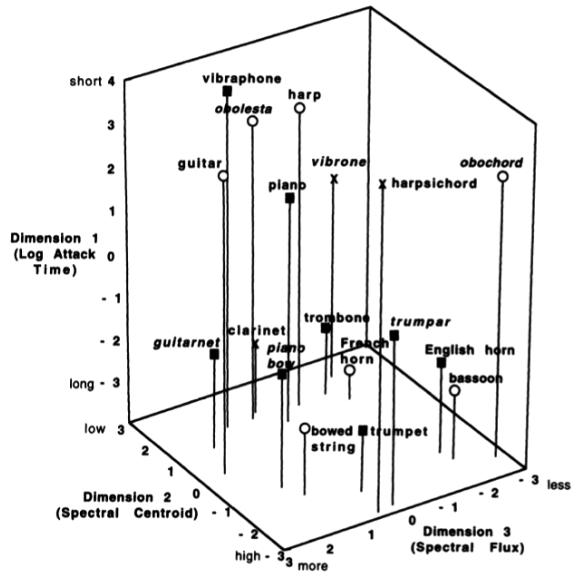


Figure 3: 3D Timbre Space from [5] Mapped to Common Orchestral Instruments

3. IMAGE SEGMENTATION

Image segmentation is a complex problem that involves separating an image into its individual components. With advancements in deep learning, many state-of-the-art solutions have been proposed. To separate an image into its salient components, this design utilizes two pretrained models: SegFormer and MiDaS. SegFormer is a semantic segmentation model, while MiDaS is a depth map model. After implementing both deep networks, the design creates masks of the important objects to be analyzed in relation to one another using their spatial attributes and corresponding depth information.

3.1. SegFormer

SegFormer consists of a hierarchical Transformer encoder and a lightweight, all-MLP decoder head. The transformers blocks are made up of self-attention layers, a

feed-forward layer, and an overlapped patch merging phase. The decoder has fully-connected layers as well as an upsampling module [6]. The model, which was pre-trained on ImageNet-1k, is a semantic segmentation model, so it is trained to segment images into like-objects. This is important for our use case because we do not care about selecting each individual tree in an image, but rather all instances of similar trees, treating them as one mask. We further trained SegFormer by fine-tuning the model on the ADE20K dataset. In our implementation, we first blurred the images with a Gaussian kernel to smooth fine details that could throw off the model stability and then convert the images into tensors using an encoder. The tensors are then used as the input to SegFormer.

Figure 4: Input image for fine-tuned SegFormer model

Figure 5: Output of fine-tuned SegFormer model

Figure 4 shows an example input image and Figure 5 displays the respective output from the fine-tuned SegFormer model. The model almost perfectly separates the grass, the trees, and the sky as three unique segments.

3.2. MiDaS

MiDaS is a large, pre-trained model that calculates a monocular depth estimation for images. The model was trained on ten different datasets, each with their own representation for depth. To solve this discrepancy, Intel's research group trained the model with predictions in disparity space as well as scale- and shift-invariant dense losses to have a depth representation that was consistent among every dataset used. One advancement that the group proposes was the use of three-dimensional movies. A large dataset consisting of thousands of frames from a variety of three-dimensional movies was constructed. Then, a stereo matching calculation was used to create the depth maps. These mappings were used as the target variable during training.

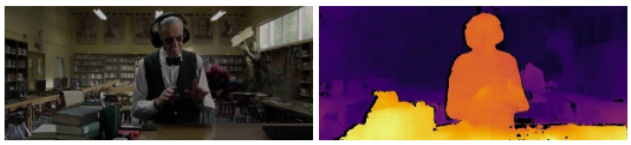


Figure 6: Three-dimensional training image

Figure 6 displays a three-dimensional image of Stan Lee with its associated calculated depth map [7].

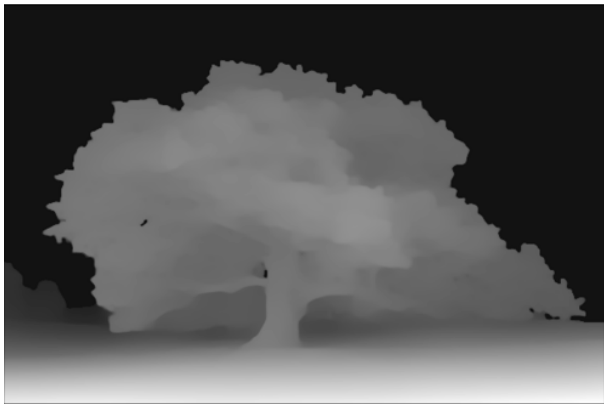


Figure 7: Output of MiDaS

An example output of MiDaS is shown in Figure 7 for the input image displayed in Figure 4. The results were exceptional, and no further fine-tuning was required for the design's implementation. In this project, the output of MiDaS is combined with the masks from the fine-tuned SegFormer model to acquire the depth information for each segmented object.

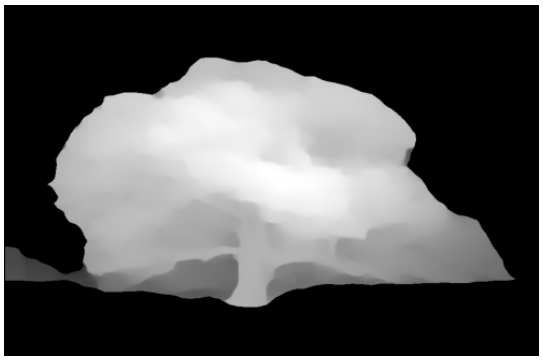


Figure 8: Masked depth map

Figure 8 displays the depth map of Figure 4 masked with the large tree segment from Figure 5. The depth information encodes the variance between gray levels in each segment, corresponding to dynamic range of the resulting voice during sonification. The grayscale histogram centroid of each mask is later calculated to find the average brightness of the segment, which translates to perceptual auditory brightness in the form of spectral centroid. The spatial centroid, or center of mass of the segment is used to determine the octave class of the generated voice corresponding to vertical placement and stereo panning mapped to horizontal placement.

3.3. Additional Proposed Methods

Many other segmentation techniques were attempted to obtain similar results. Some of these methods include DeepLab, Segment Anything Model (SAM), canny edge detection, Gabor filtering with k-means clustering, and a k-nearest neighbor algorithm. DeepLab is another pre-trained semantic segmentation model using a different method compared to SegFormer [8]. This was not used because SegFormer is much easier to implement and fine-tune and yielded better results for this design's needs. SAM, another pre-trained deep model for image segmentation, gave even worse results, as the model attempted to separate far too many fine details in the images [9]. Canny edge detection was also used. The team thought that applying the algorithm to the results of MiDaS and thresholding would result in obtaining objects at different locations. This worked, however it got rid of the semantic information. Therefore, multiples of the same type of object were separated, which is not what the team desired. Gabor filtering with k-means clustering and k-nearest neighbor algorithms were both great options as they performed textural separation well, a key component of the process. However, they suffered from one crucial flaw: the number of important objects in each image is unknown. Therefore, these effective tools did not apply to the project.

4. TEXTURE AND COLOR ANALYSIS

The texture analysis consists of image processing algorithms used to form a normalized timbre "coordinate" which is taken by the music generation process as an input. Color analysis involves extracting the most frequent colors in each image segment and then assigning them to a key based on the previously defined color relations in the circle of fifths.

4.1. Histogram Centroid

The image luminosity histogram centroid is chosen to create a normalized value from 0 to 1 corresponding to the spectral centroid. The motivation behind this is that image brightness scales linearly with grayscale luminosity value, and auditory brightness scales linearly with frequency. This corresponds to a common association between bright and dark sounds to bright and dark images and is consistent with Tanay's synesthetic mapping. For image grayscale histogram $h(p_i)$ where p_i is the i^{th} pixel of the NxM converted grayscale image with values ranging from 0 to 1, we compute

$$Histogram\ centroid = \frac{\frac{1}{NM}\sum_{i=1}^{NM} p_i h(p_i)}{\sum_{i=1}^{NM} h(p_i)}$$

creating a value between 0 and 1. This is later scaled to match the audible frequency range.

4.2. Neighborhood Gray Tone Differences Matrix (NGTDM)

The Neighborhood Gray Tone Differences Matrix (NGTDM) "quantifies the difference between a gray value and the average gray value of its neighbours within distance δ", which corresponds to spatially varying visual texture details [10]. From the matrix, five features can be extracted to quantify perceptual aspects of the visual texture. The formulations of each are outlined below where $N_g = 256$ denotes the number of gray levels, p_i denotes gray level probability, and s_i the sum of absolute differences for gray level i in the resulting matrix.

Coarseness:

$$\frac{1}{\sum_{i=1}^{Ng} p_i s_i}$$

Contrast:

$$\left(\frac{1}{N_{g,p}(N_{g,p}-1)}\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p_i p_j (i-j)^2\right)\left(\frac{1}{N_{v,p}}\sum_{i=1}^{N_g} s_i\right), \text{where } p_i \neq 0, p_j \neq 0$$

Busyness:

$$\frac{\sum_{i=1}^{Ng} p_i s_i}{\sum_{i=1}^{Ng}\sum_{j=1}^{Ng} |ip_i - jp_j|}, \text{where } p_i \neq 0, p_j \neq 0$$

Complexity:

$$\frac{1}{N_{v,p}}\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} |i-j|\frac{p_i s_i + p_j s_j}{p_i + p_j}, \text{where } p_i \neq 0, p_j \neq 0$$

Strength:

$$\frac{\sum_{i=1}^{Ng}\sum_{j=1}^{Ng}(p_i+p_j)(i-j)^2}{\sum_{i=1}^{Ng} s_i}, \text{where } p_i \neq 0, p_j \neq 0$$

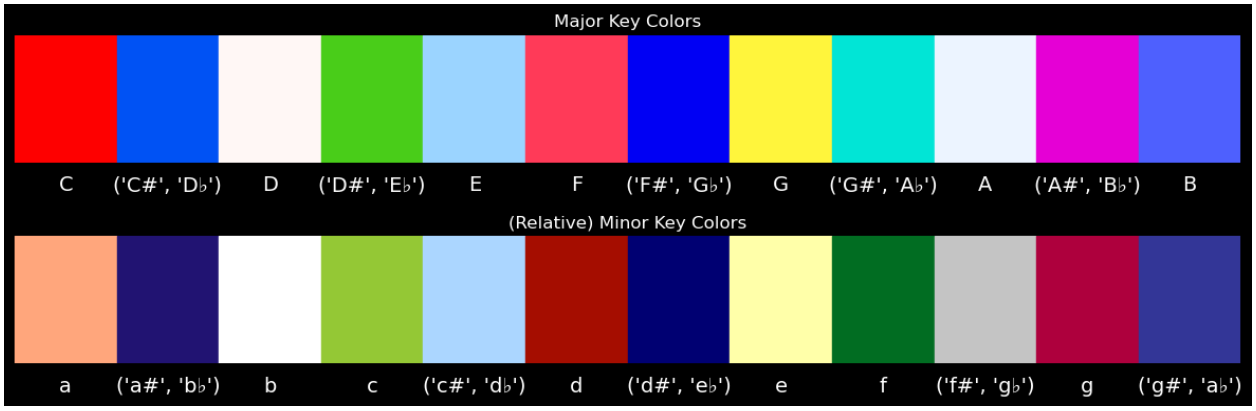


Figure 9: Color Dictionary

We choose to map coarseness, busyness, and complexity to spectral flux because it quantifies how rough something sounds at a given time. Contrast and strength are inversely proportional to log attack time as these qualities represent local visual dynamic range, intuitively corresponding to how quickly the sound starts, or its attack. These values are scaled, averaged and roughly normalized from to be modified in the audio synthesis stage.

4.3. Color-Key Dictionary

The key-to-color relations are constructed as a dictionary in Python and later enumerated as data points. Figure 9 is derived as a linear chromatic realization of Figure 1 to highlight the similarities and color shifts present.

4.4. Perceptual Color Distance

Based on the reasoning mentioned in Section 1, the Hue Saturation Value representation of color shown in Figure 10 is used in tandem with the RGB space.

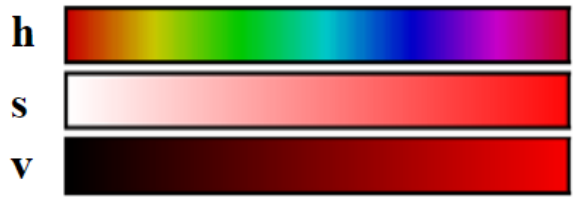


Figure 10: HSV Color Space

From the r, g, and b values in the RGB color space and the h, s values in the HSV space, we derive the following perceptual color distances.

$$d_{rgb} = \sqrt{(r_{out} - r_{in})^2 + (g_{out} - g_{in})^2 + (b_{out} - b_{in})^2}$$

$$d_{hsv} = \sqrt{(h_{out} - h_{in})^2 + \frac{1}{2}(s_{out} - s_{in})^2}$$

RGB distance d_{rgb} is simply the Euclidean distance between two RGB colors, while d_{hsv} or HSV distance is weighted to favor hue over saturation, discarding value entirely as this is tied to brightness which is accounted for in the histogram centroid. From these values, the closest color in the dictionary is found by minimizing the total distance from an observed color's hex value to any hex value in the dictionary as follows.

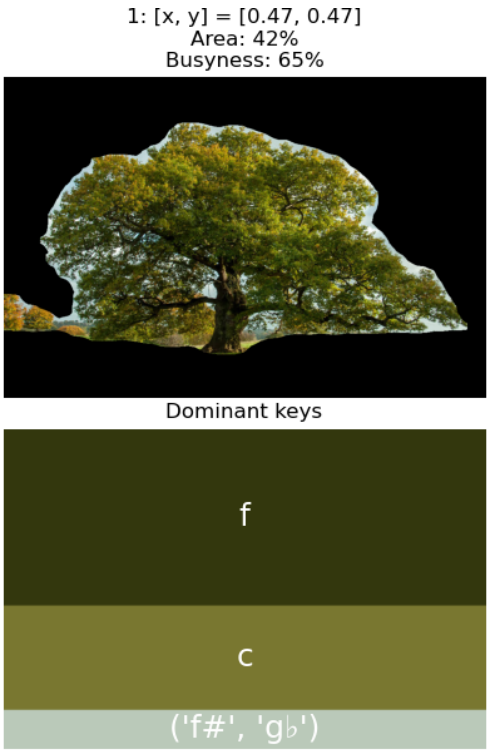
$$c_{out} = \underset{c_{dict}}{\operatorname{argmin}}(d_{rgb} + d_{csv})$$

4.5 K-Means Color Segmentation

The k-means algorithm clusters RGB points in 3D color space based on distance to find most frequent colors by segment in an image. This essentially performs a version of RGB compression and greatly simplifies the number of possible colors to consider for the distance calculation. We use the results to create a color palette for each segment, thus finding the most probable keys.

4.6 Analysis Results

The results of segmentation and image analysis are shown in Figure 11 for color and spatial attributes and Figure 12 for textural attributes once mapped to their defined auditory equivalents. An example is shown for the image shown in Figure 4.



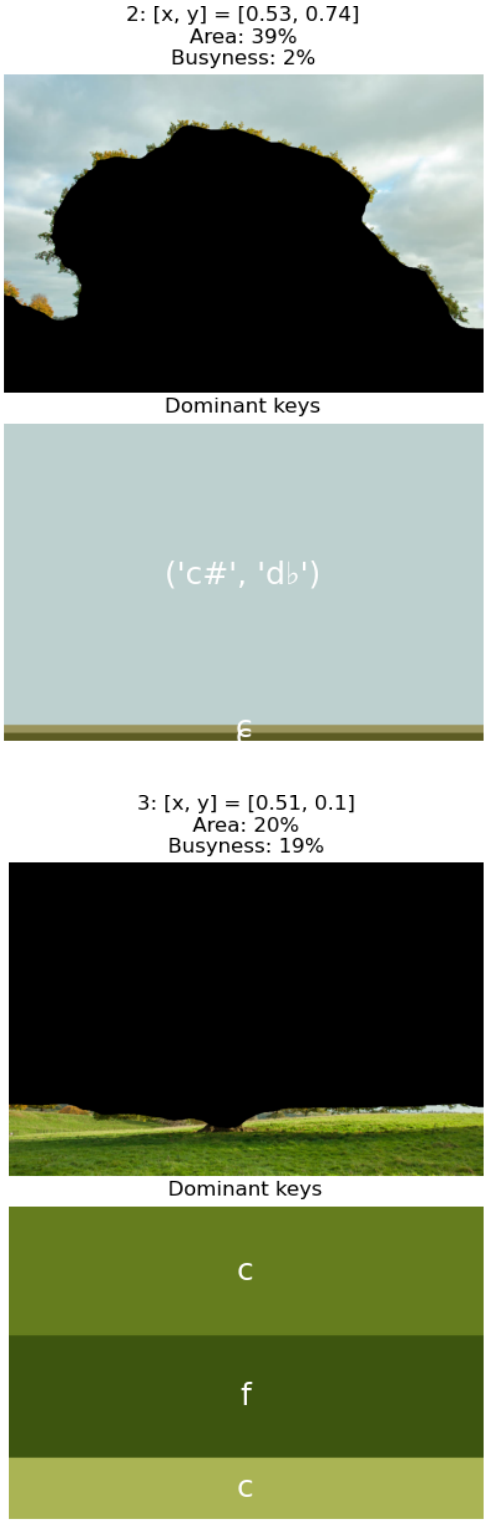


Figure 11: Segmented Color and Spatial Features

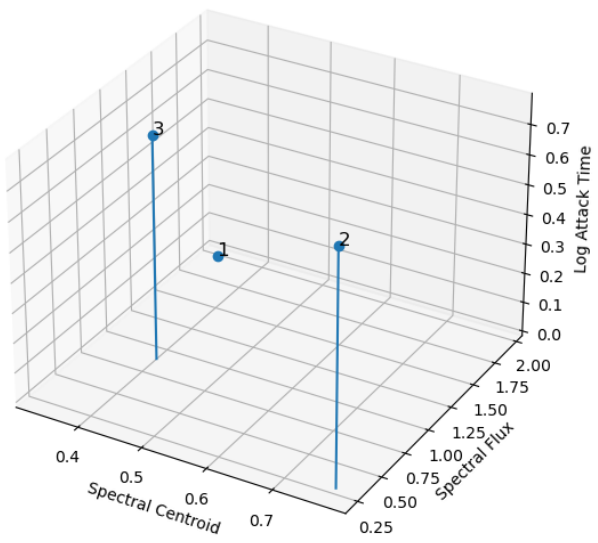


Figure 12: Segmented Timbral Attributes

4.7. Additional Proposed Methods

For the analysis of visual roughness, properties extracted from the Gray-Level Co-occurrence Matrix (GLCM) were also considered, but proved to be inconsistent with the overall perception of roughness in the image for this application. Histogram Spread was also considered as a feature mapped to log attack time because it maps the overall sense of image contrast as a value from 0 to 1 [11]. However, the contrast and strength features from the NGTDM proved more effective for this step. Additionally, higher-level texture descriptors from the Describable Textures Dataset were considered as classifiers resulting from a Wavelet Scattering deep neural network with the desired image as input [12]. Although the model was successfully trained and run, the implementation proved to be difficult due to the abstract nature of the descriptors and lack of interpretability in mapping them to synthetic audio textures.

5. AUDIO SYNTHESIS IN Max/MSP

Max/MSP, a graphical programming language designed for A/V processing, was used to generate the resulting audio.

5.1. Composition

The main compositional object is a minimalism object in Max/MSP [13], referred to hereafter as a "voice" which is modified to take input from a text file as parameters for composition. One voice is chosen as the "master" which controls the metronome and transitions for all other objects. Tables within a voice object act as banks of notes and were generated as constants beforehand, containing notes for each possible key. A window of randomly selected (constrained) size is chosen at the beginning of each phrase, containing note information that makes up a melodic "pattern". The pattern is repeated throughout the course of the phrase, with image busyness affecting how often the "base note", or first note, is repeated. This results in a more repetitive pattern at lower busyness levels, instead of flowing melodic patterns. Additionally, a drone object is included to provide a base (bass) note for the master voice, helping strengthen tonality by including a pedal tone to imply a chord on each melodic pattern.

5.2. Textural Data

Spectral centroid, spectral flux, and log attack time were translated to simpler harmonic changes in order to preserve consonance and reduce program complexity. A set of notch filters are applied to the output in cascade, with their center

frequencies updating in real-time to odd-harmonic frequencies of the voice. Spectral centroid is normalized and is used as the Q-factor for each of these filters. Different waveforms with increased harmonic complexity were chosen with spectral flux as an input parameter. Attack time is scaled and passed to the adsr~ object to modify the attack time of each voice.

5.3. Other Perceptual Information

Color is enumerated and translated to Tanay's corresponding keys. These values appear in each line of the text file with image data in it. These colors are passed directly as keys for modulation at the end of a phrase. X/Y position of an object within the image (based on the object's centroid) corresponds to L/R panning and octave class, respectively. Object size determines the length of each phrase, or the time before modulation. When modulation occurs, data from the next object is passed to the program, and a new set of melodic patterns in the new key is generated by the voices.

6. CONCLUSION

SSIMAGESS accurately depicts the inverse of what a synesthete perceives. Instead of seeing visual representations when a song is played, this process generates minimalist music based on both Tanay's synesthetic correspondences and calculated information from the images themselves. Multiple phases of the musical output are composed from the different segments of the image. Future work that the team plans to accomplish is adapting this procedure to generate scores for films. A possible solution for this would be to generate a music clip for each shot in a movie, and the image chosen would be from pertinent frames in that shot. We also expand to improve on the musicality of the generated audio and extend the idea to video segments instead of static images. A link is provided to watch a condensed and entertaining video summary of this article.

REFERENCES

[1] Brang, David, and V.S. Ramachandran, "Survival of the synesthesia gene: why do people hear colors and taste words?," *PLoS biology*, vol. 9, no. 11, 2011.

[2] Cianci, Lisa, "Additive and Subtractive Colours on the Colour Wheel," *Rmit.pressbooks.pub*, 23 Feb. 2023.

[3] Berger, Kenneth W., "Some Factors in the Recognition of Timbre." *Journal of the Acoustical Society of America*, vol. 36, no. 10, pp. 1888-91, 1964.

[4] Mcadams, Stephen, "The Perceptual Representation of Timbre," 2019.

[5] McAdams, Stephen, "Perspectives on the Contribution of Timbre to Musical Structure," *Computer Music Journal*, vol. 23, no. 3, pp. 85–102, 1999.

[6] Enze Xie, undefined., et al. "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers," in CoRR, vol. abs/2105.15203, 2021.

[7] Katrin Lasinger, undefined., et al. "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot

Cross-Dataset Transfer," in CoRR, vol. abs/1907.01341, 2019

[8] Liang-Chieh Chen, undefined., et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," in CoRR, vol. abs/1606.00915, 2016.

[9] Alexander Kirillov., et al. "Segment Anything," 2023.

[10] M. Amadasun, R. King, “Textural features corresponding to textural properties,” *Systems, Man and Cybernetics, IEEE Transactions*, vol. 19, pp. 1264-74, 1989.

[11] A. K. Tripathi, S. Mukhopadhyay and A. K. Dhara, "Performance metrics for image contrast," *2011 International Conference on Image Information Processing*, Shimla, India, pp. 1-4, 2011.

[12] “Describable Textures Dataset,” *robots.ox.ac.uk*, www.robots.ox.ac.uk/~vgg/data/dtd/.

[13] “Minimal loops using a portion of a table | Max Cookbook,” *music.arts.uci.edu*. https://music.arts.uci.edu/dobrian/maxcookbook/minimal-loops-using-portion-table