THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF ELECTRICAL ENGINEERING

INTUITIVE MAPPING OF WALLPAPER SYMMETRY GROUPS TO MUSIC SYNTHESIS

ANDY LUO
SPRING 2019

A thesis
submitted in partial fulfillment
of the requirements
for baccalaureate degrees
in Electrical Engineering and Computer Science
with honors in Electrical Engineering

Reviewed and approved* by the following:

Yanxi Liu
Professor of Computer Science and Electrical Engineering
Thesis Supervisor

Julio Urbina
Associate Professor of Electrical Engineering
Honors Adviser

*Signatures are on file in the Schreyer Honors College.

ABSTRACT

Symmetry exists all around us in our natural world and embeds a wealth of valuable information to be interpreted. Human brains are hardwired to seek out and catalog symmetries in order to distill the massive quantity of visual signals constantly being received. Group theory and computer vision have already been used in conjunction to detect and interpret various visual symmetries found naturally and synthetically, bringing machine perception closer to human perception. This work examines the possibility of intuitive mapping of symmetry groups from visual to auditory domain, specifically to music. Secondary visual characteristics such as color, shapeliness, and frequency, among others, are also mapped to music generation parameters. The effectiveness of mapping 2D visual images, especially its symmetry structure and texture, to sound was demonstrated. Various types of audio mappings were devised and tested, with the goal of maximizing discernment among different symmetry groups and also secondary characteristics within symmetry groups. These mappings were implemented in real-time using a synthetic image set. Specifically, a melody motif-based mapping was developed and achieved better discernibility through listening tests than baseline mappings. This experimental method suggests some degree of linkage between visual and auditory symmetry pattern perception in the brain that can be explored through future work.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Huge thank you to Dr. Liu for encouraging me to explore this experimental idea and entrusting me with the project on a short notice. Thank you to my parents and close friends for the never-ending support. Everyone who helped with data collection played an integral part in the completion of this thesis. Finally, thank you Gourdo, my pet gourd.

Chapter 1

Introduction

1.1 Symmetry

Symmetry exists in virtually all aspects of the natural world, from organic structures to galaxies. Human perception relies heavily on pattern recognition and recall. With the sheer magnitude of sensory input entering the brain, pattern recognition is a key way the brain can perform information abstraction and formulate higher understanding. The human brain discerns symmetry features preattentively, suggesting that the information embedded by symmetry holds significant value for higher level perception functions, such as object recognition.

Although different branches of mathematics will have their own formal definitions of symmetry, the one most relevant to perception is as follows: Given a structured object, a symmetry is a transformation on that object which maps it back to itself and preserves the structure. Weyl describes the idea as starting with the vague notion of symmetry equaling harmony of proportion and expanding it to say that symmetry is the invariance of elements under a group of automorphic transformations [1]. Informally, a symmetric object is one that is made up of smaller interchangeable objects.

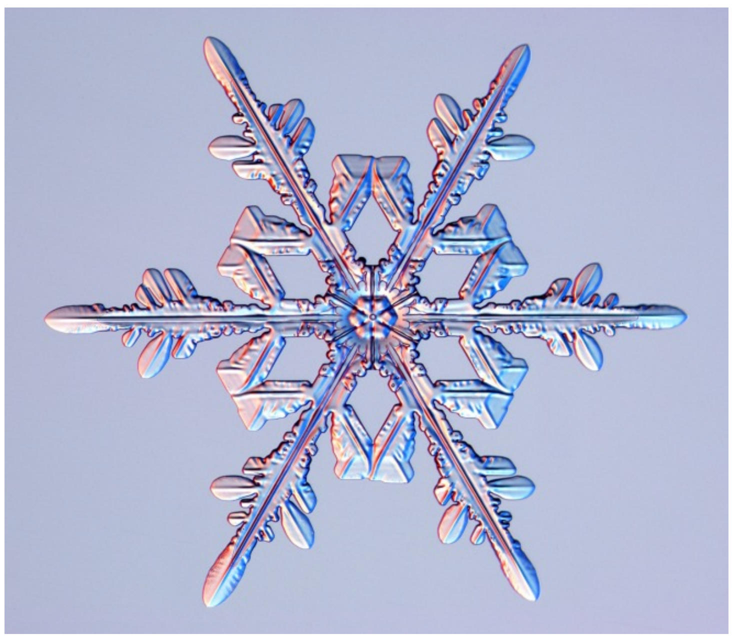


Figure 1.1: Snowflakes: a real world example of objects that contain similar symmetry rules but unique instantiations.

The human senses of sight, hearing, and touch all distinguish symmetries and contribute to the overall goal of object/entity recognition. Although details regarding the neurological pathways that facilitate these functions are beyond the scope of this thesis, there is interest in whether any level of connection exists between symmetries of different senses. Through functional magnetic resonance imaging, it has been determined that visual symmetries are processed in the occipital lobe and register as quickly as 50 milliseconds after stimuli presentation [2]. However, there exists little to no prior work on whether visual and auditory symmetries share any amount of neurological common space, and this thesis aims to explore through experimental means if any value can be derived from a system that utilizes both.

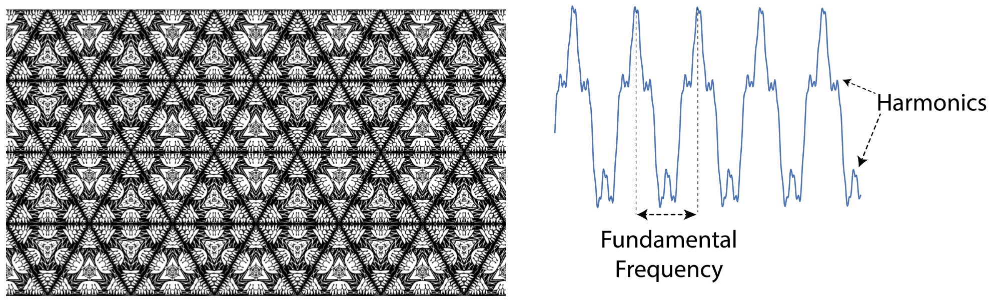


Figure 1.2: Examples of visual (left) and auditory (right) symmetries

1.2 Wallpaper Symmetry Groups

Liu provides a formal definition of spatial symmetry [3]:

> More formally, in a metric space M, a symmetry $g \in G$ of a set $S \subseteq M$ is an isometry (a distance preserving transformation) that maps S to itself (an automorphism), $g(S) = S$. The transformation g keeps S invariant as a whole while permuting its parts. Symmetries G of S form a mathematical group $\{G, *\}$, closed under the transformation composition $*$, called the *symmetry group* of S.

Additionally, for all periodic patterns in R^n for any n, it was discovered that there is only a relatively small finite number of symmetry groups, referred to as *crystallographic groups* [3]. In two-dimensional (2D) euclidean space, translation, rotation, reflection, and glide-reflection serve as the primitive symmetries. In conjunction, they build the seventeen symmetry groups of 2D space often labeled as the wallpaper symmetry groups. Any instance of symmetry in 2D space (of which there are infinite) belongs to one of the seventeen wallpaper groups.

1.2.1 Wallpaper Patterns

(a) CM (b) P6 (c) PM

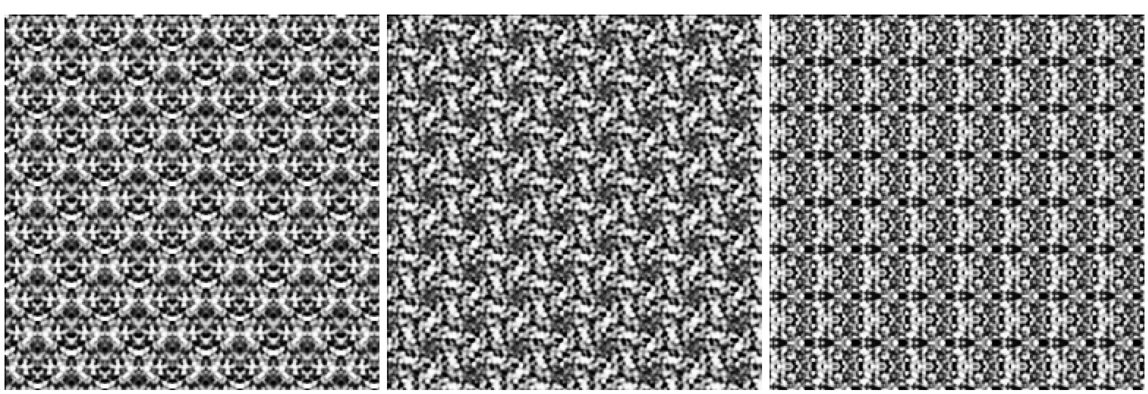


Figure 1.3: Example synthesized patterns from the CM (left), P6 (middle), and PM (right) wallpaper groups

All translationally symmetric patterns can be generated by a pair of vectors t_1 and t_2, which are linearly independent and shortest among all possible vectors. These two vectors define the shape, size, and orientation of a unit lattice tile that can span the entire pattern space. Certain pairs of vectors define the true underlying symmetry better than others. If rotation symmetries are present, tiles centered on rotation centers usually reflect the symmetry most completely. As defined by Liu, ”A motif of a wallpaper pattern is a tile that is cut out by a lattice whose unit is centered on the fixed point of the largest stabilizer group”, where stabilizer group here is a symmetry group that leaves a single point invariant [3].

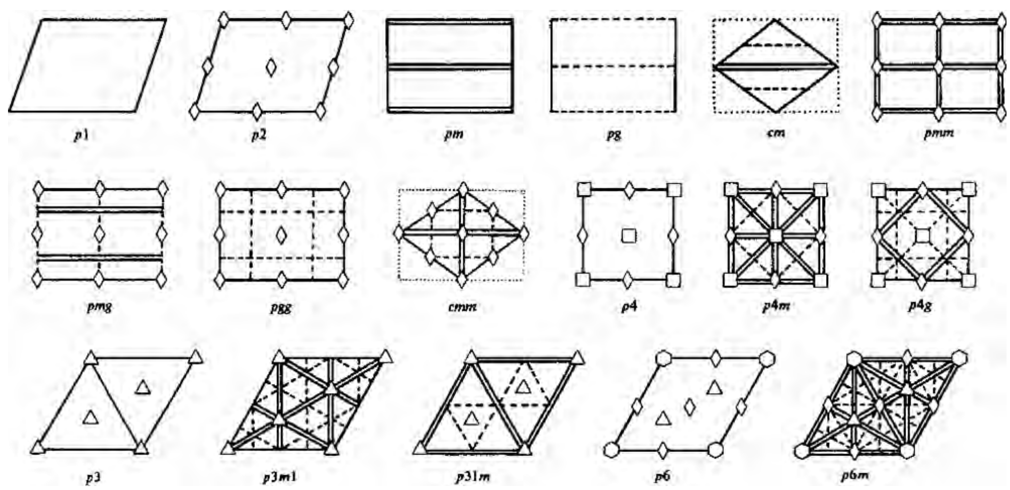


Figure 1.4: The unit lattice for each of the seventeen wallpaper groups, where the diamond, triangle, square, and hexagon icons indicate the locations of 2-,3-,4-, and 6-fold rotation centers, respectively.

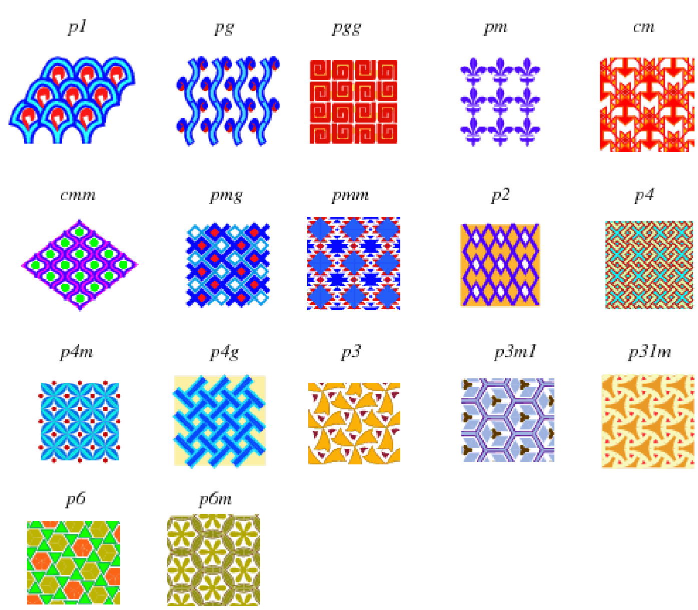


Figure 1.5: An example wallpaper pattern from each of the seventeen wallpaper group, courtesy of [4]

1.2.2 Intra-Group Characteristics

Although there is a finite number of symmetry groups, there are an infinite number of possible patterns of each symmetry group, with potential differences in texture, shape, color, just to name a few characteristics. Since a wallpaper pattern is the repeated tiling of a unit lattice in 2-dimensions, the unit lattice itself contains all of the defining characteristics of that specific pattern.

1.3 Musical Symmetry

Music can also be thought of a collection of symmetries (or near symmetries) that exist along the time axis. A typical music track has rhythm and melodies arranged into repeated blocks, played by a collection of real and virtual instruments. Auditory symmetries are defined by amplitude over time. Musical symmetry exists in a space defined by sounds over time, where the sounds themselves are composed of auditory symmetries. Layered auditory symmetries ultimately dictate the timbre of a sound, while layered musical symmetries form the structure of a piece of music. Although both are vital to how a piece of music sounds, they can be independently described (acoustic characterization of violin sound vs. violin sheet music). Auditory symmetries contribute to the brains ability to distinctly identify and catalog different sounds and form higher level associations. Musical symmetries support the cohesiveness of a single musical work and assists the brain in compressing/remembering musical ideas.

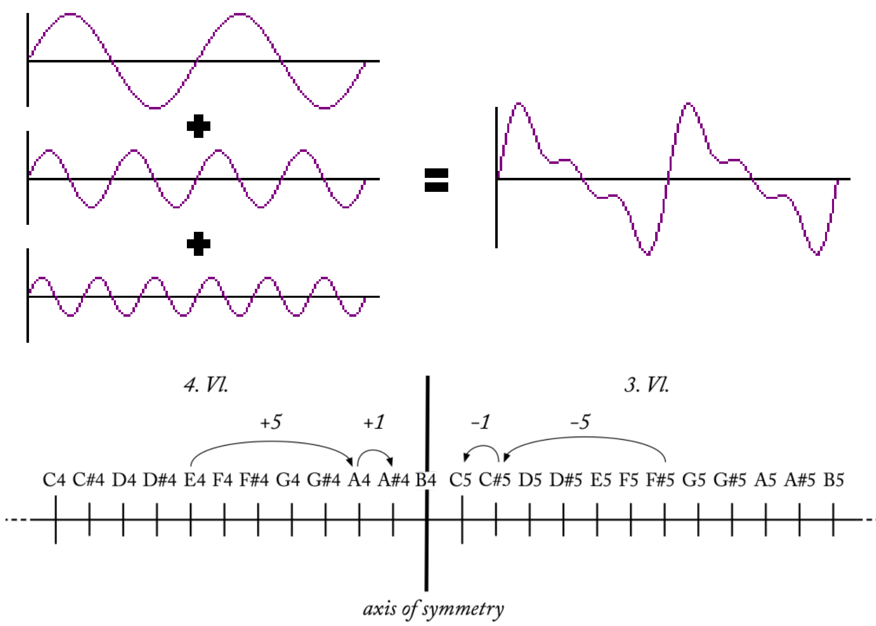


Figure 1.6: The symmetry present in waveforms of individual sounds (top), courtesy of [5], and the symmetry present in music structure (bottom), courtesy of [6]. This is a specific example of melodic pitch symmetry.

1.4 Musical Structure

Modern music has become quite similar in song structure, even across multiple genres. Instead of using seconds, the timeline of a track uses beats as the base unit of subdivision. Usually, the percussion parts will establish the tempo and the underlying heartbeat of the track, giving the listener a clear sense for the equally spaced beats. Every four beats are grouped together to form a measure. Every eight or sixteen measures will form a phrase, the musical equivalent of a sentence. Usually, a complete melodic idea is expressed within a phrase. A self-contained track will group phrases into different sections that vary in energy and feel. Some common groups of phrases include the verse, which roughly corresponds to a poetic stanza, and the chorus, which is a section of the music that is often high energy, familiar, and gives a sense of return. A chord progression usually spans an entire phrase and gives the song a chromatic foundation, with each chord within the chord progression lasting one or two phrases. A song may contain a handful of different repeated chord progressions. From the percussion, to melodies, to chords, there is a high level of translational symmetry at every level of abstraction within a song.

Chapter 2

Approach

2.1 Wallpaper Symmetry Group Synthesizer

The system presented by this thesis takes a visual input belonging to a wallpaper symmetry group and generates music accordingly, capturing as much detail regarding to the input as possible. Dubbed WallpaperSynth, this prototype system is comprised of two parts: a module that performs the actual music generation given a visual input, and a means for performing human assessment on the effectiveness of the visual to musical mapping. Since intuitiveness is a subjective measurement, the main assessment is how well a user can select the correct input given a generated song, which provides some quantitative measure of mapping effectiveness. To narrow the scope of implementation, the system only includes a subset of the seventeen 2D wallpaper symmetry groups, but can be scaled up to cover all seventeen. This system is ported as both a local application, for integration with other systems, and a web application, for ease of data collection.

The music generation module has a straightforward pipeline. Given an input image which contains an instance of a wallpaper symmetry group, a routine classifies the symmetry group. This is done using the lattice detection technique as described by Liu in [3]. Additionally, the secondary visual characteristics are analyzed and this set of parameters are passed into the synthesis routine. The synthesis process uses the determined symmetry group for song structure generation and the secondary parameters for tonal formation and modification. The music generation happens continuously in real-time, meaning that for any sequence of inputs, the music remains continuous and transitions on-rhythm between the corresponding melody sequences. As long as there is an input present, the module will output music in the form of one continuous song.

2.2 Input Images

For both development and human assessment, the same synthetic wallpaper pattern imageset is used. This imageset spans all seventeen symmetry groups and contains 500 wallpaper images per symmetry group, 8,500 in total. Each image is tiled from a randomly generated grayscale motif and has 256x256 pixels. Examples of the images can be seen in Figure 1.3. The entire imageset is normalized for better classifier performance. Additionally, each image is filtered with a 3x3 pixel Gaussian and histogram equalized.

2.2.1 Secondary Characteristic Augmentation

All the images within each wallpaper group are visibly different. Frequency analysis of the images reveals that all of the images have the same narrow bands of frequencies (and their multiples), as evident in Figure 2.1. The dominant frequency components come from the the size of the unit lattice, not the contents of the lattice itself. The entire imageset is grayscale so for development involving color mappings, color features are augmented in real-time pre-processing.

(a) P1 Pattern (b) FFT

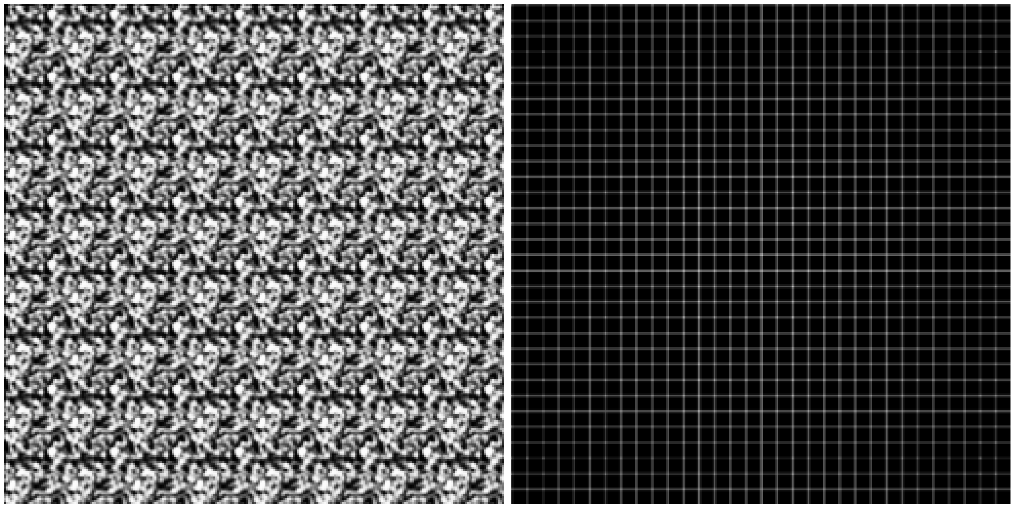


Figure 2.1: Frequency analysis of wallpaper pattern from the P1 group. Right is pattern, left is FFT of the pattern

2.3 Base Mapping

In order to establish baseline performance, a naive symmetry group to musical structure mapping was conceived. This mapping, which covers seven of the seventeen wallpaper groups, simply assigns a note to each of the groups. These notes span a single octave from C4 to B4 in the key of C Major. No secondary visual characteristics are mapped and the tonal qualities of the sound are instead also mapped from the symmetry group. Each group is assigned a different synthesizer, with sounds ranging from plucks to detuned jabs. There are two versions of the base mapping, one for the local application and one for the web application, which only differ in synthesizer sounds because the two environments have different music coding libraries.

Wallpaper Group	**Note**	**Synthesizer**	**Synthesizer Description**
P1	C4	pretty-bell	Pretty bell sound
P2	D4	tri	A simple triangle wave
PM	E4	hoover	Classic early 90's rave synth
PG	F4	hollow	A hollow breathy sound (from noise)
CM	G4	piano	A basic piano sound
PMM	A4	pluck	A Karplus-Strong synthesized pluck
PMG	B4	zawa	Saw wave with oscillating timbre

Table 2.1: Overview of the base mapping for local application (base mapping v1)

Wallpaper Group	Note	Synthesizer	Synthesizer Description
P1	C4	Synth	A simple oscillator
P2	D4	AMSynth	A simple amplitude modulation synth
PM	E4	DuoSynth	Two parallel synths linked by frequency ratio
PG	F4	FMSynth	A simple frequency modulation synth
CM	G4	MetalSynth	A inharmonic and spectrally complex source
PMM	A4	PluckSynth	A Karplus-Strong synthesized pluck
PMG	B4	MembraneSynth	Kick and tom drum sounds

Table 2.2: Overview of the base mapping for web application (base mapping v2)

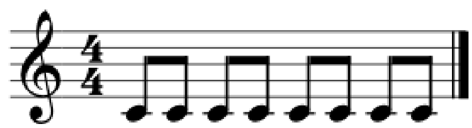


Figure 2.2: The rhythm for all melodies of both versions of the base mapping. The note pitch value corresponds to the P1 group.

The sound is played as a repeated 8th note in a 4/4 time signature. Due to the arbitrary nature of this mapping, it is hypothesized to be difficult for a user to form correspondences. Even consciously, perfect pitch is required to discern the differences between the notes, and more consequentially by extension, the symmetry groups they were mapped from.

2.4 Musical Motif-Based Mapping

Instead of mapping each symmetry group to a repeated note, each symmetry group is represented by a measure-long musical motif. This motif-based mapping leverages the fact that a short melodic word can contain symmetries that provide an intuitive correspondence to a visual symmetry group. Five out of the seventeen wallpaper groups are included with this mapping. Groups with differing maximum rotation center magnitudes are chosen for this mapping, as this improves perceptive separability between the groups, and gives more flexibility in musical motif representation. This mapping tests the perception of musical symmetries and whether they can intuitively correspond to visual symmetries. To isolate the effects of melodic and rhythmic symmetry from timbre, all musical motifs are played using the same synthesizer sound, a simple sine tone.

Wallpaper Group	Maximum Rotation Center	Dominant Notes
P1	2-fold	8th and dotted-quarter notes
P2	2-fold	8th and dotted-quarter notes
P3	3-fold	Half note triplets
P4	4-fold	Quarter notes
P6	6-fold	Quarter note triplets

Table 2.3: Overview of motif-based mapping and the dominant note type for the subset of five symmetry groups.

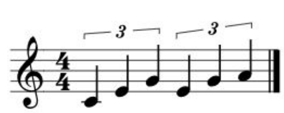
Wallpaper Group	Repeated Melodic Motif
P1	
P2	
P3	
P4	
P6	

Table 2.4: The repeated measure-long melodies of motif mapping v1

Each of the melodies are hand-designed, and not generated procedurally from the geometry of the wallpaper group. Since wallpaper groups are defined in 2D Euclidean space and musical symmetries are defined in auditory events over time (1D), there is an inherent spatial mismatch that inhibits the creation of a direct mathematical mapping. Instead, the motifs are created subjectively by examining the perceived effect of the maximum magnitude rotation center for each wallpaper group, in essence finding a combination pitch and rhythm sequence that alludes to the geometry of the visual symmetry. For example, the P6 group has 6-fold rotation centers which makes its hexagonal symmetry stand out. The hexagon shape, with its six equal sides, is evocative of quarter note triplets, which splits a measure into six equal time divisions. The triplets sound distinct since the underlying times signature is in four. The P4 group on the other hand has 4-fold rotation centers and gives the patterns an overall square structure. This intuitively alludes to a melody with quarter note rhythm. The other musical motifs are created using similar design rationales.

2.5 Accompaniment

For the generated song to sound realistic, accompanying parts are played as a backing for the mapped melody. The accompaniment loops constantly regardless of melody changes. This gives the generated song a continuous cohesiveness, which should sound pleasant for the listener.

2.5.1 Chords

A chord progression is used to establish phrasing in the generated song and gives the track a harmonic foundation. Since all the generated melodies are in the key of C major, a common 4-part chord progression is used: Cmaj, Amin, Dmin, Fmaj. Each chord lasts for the duration of a measure and the entire sequence loops indefinitely.

Figure 2.3: Musical notation of the accompanying chord progression (Cmaj, Amin, Dmin, Fmaj).

The local application uses the blade synthesizer, which produces a 80’s style synth lead sound, while the web application uses a 6-voice polyphonic amplitude modulation synthesizer that has a slightly more airy sound than the blade synthesizer. The chords are mixed at a quarter volume of the main melody.

2.5.2 Percussion

In order to keep the pulse of the track, a simple drum pattern is implemented with kick, snare, hi-hat, and rim sounds. The percussion part is especially crucial for distinguishing between four-beat and triplet melody rhythms. A very typical measure-long pattern is implemented with no complicated rhythms.

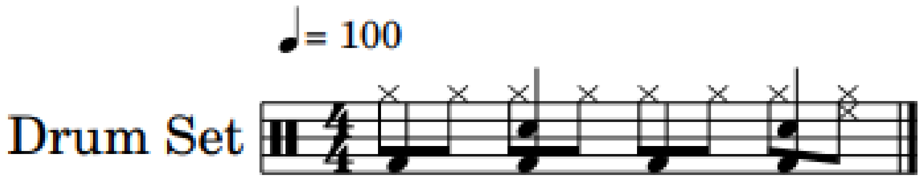


Figure 2.4: Musical notation of the accompanying percussion part, a four on the floor pattern.

Having a kick drum, the most prominent percussion sound, on each beat gives the track non-distracting yet strong rhythmic foundation. This four on the floor pattern is very popular with electronic music, which fits the overall synthesized nature of the track. Snare sounds are present on the 2nd and 4th beats. 8th note hi-hats are present along with a rim hit on the last 8th note of the measure. The sounds all come from the iconic 808 synthesizer, which is accurately emulated on both the local and web applications. The 808 sounds are ubiquitous to the world of electronic music and provides a non-distracting tone.

2.6 Secondary Parameter Mapping

Secondary visual characteristics, such as texture and color, also have a large influence on the way an image is perceived. There is value in exploring how these characteristics, which are continuous relative to discrete wallpaper groups, can map to an auditory parameter.

2.6.1 Texture

In the case of wallpaper patterns, a texture is defined by the size and shape of the repeated motif as well as the actual design of the motif. To better explore the textural impact of the motif design itself, all of the wallpaper patterns in the input imageset have roughly the same size and shape unit lattices. This isolates the effect of the pattern from the effect of the scale.

Gray-level co-occurence matrix (GLCM) is a statistical technique for quantifying texture characteristcs such as entropy, contrast, and homogeneity [7]. The technique can be thought of building up joint probability distribution P, where $P[x_1, x_2]$ represents the number of times a pixel has gray level x_1 and a pixel that is a fixed displacement away has the gray level x_2. Essentially, this is measuring the gray levels separated by a fixed displacement throughout the entire image.

For a given input image I and displacement vector $d = [d_x, d_y]$, P is built up in the following manner:

1. Scan the image row by row from the top left to bottom right.
2. Given the current pixel (I_x, I_y) has a gray level of m and the pixel (I_{x+d_x}, I_{y+d_y}) has a gray level of n, increment $P[m, n]$ by one.
3. Repeat for all pixels in I.

The GLCM P is a MxM matrix where M is the number of gray levels in the image. The input images are all 8-bit grayscale images which is 256 gray levels. To distill the texture information into a smaller GLCM, the images are re-quantized on the fly to 4 bits, which results in a 16 by 16 GLCM. The following metrics can be determined once P is estimated:

- Entropy - has maximum value when P is uniform (random texture) and minimum value when distribution is deterministic (uniform texture).

$$Entropy = -\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} P[i,j]log_2 P[i,j] \tag{2.1}$$

- Energy - has smallest value when all values in P are the same (random texture) and maximum value if P is only populated in one cell.

$$Energy = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} P[i,j]^2 \tag{2.2}$$

- Contrast - low in magnitude if the values along the diagonal of P are large (regions of uniform gray levels), and large in magnitude otherwise.

$$Contrast = \sum_{i=0}^{M-1}\sum_{j=0}^{M-1} (i-j)^2.P[i,j] \tag{2.3}$$

- Homogeneity - Opposite of contrast, high value when P mostly populated along diagonal and low in value otherwise.

$$Homogeneity = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{M-1} P[i,j]}{1+ \mid i-j \mid} \tag{2.4}$$

Entropy and contrast are used as inputs for the music generation module. Entropy is mapped to a flanger effect node and contrast is mapped to a reverb effect node, in serial. These nodes are placed after the base song generation, which means all sounds pass through both modules. The magnitudes of the metrics are directly mapped to the dry/wet ratio of the effect, which determines the relative levels of dry (un-effected) and wet (effected) signals. A higher contrast means more randomness and a strong flanger effect subjectively gives off the impression of randomness. A higher contrast means smaller uniform gray level areas, which is subjectively alluded to by high reverberations. These mapping decisions are have no mathematical basis and act as an experimental starting point.

2.6.2 Color

Although the perception of color is complex and nonlinear, a rudimentary mapping may provide some degree of color discernment through auditory means. For an input image, the average intensities of the red, green, and blue channels are measured, and mapped to a 3-part oscillator. This oscillator synthesizer plays the same notes of the chords and contains a sine wave, saw tooth wave, and square wave. The relative mixed levels of these sub-oscillators are determined by the relative ratio of color intensities. Since color is largely separable from texture and structure visually, it is mapped to a tonal quality of the sound that doesn't interfere with the main melody or chord progression. It simply changes the timbre of one accompaniment part.

2.7 Listening Tests

In order to test the discernment of wallpaper groups from the above mappings, a series of audio-visual listening tests are devised. These tests ask the user to choose the wallpaper patterns that corresponds to the generated songs. These tests are distributed through the web application which allows a broad collection of people to be reached, with varying levels of musicality.

Before the test is administered, the user is instructed to spend roughly five to ten minutes with the WallpaperSynth player, which randomly loops through different wallpaper patterns and plays the continuously generated song in response. The user controls when the player switches to the next random pattern so they can spend as much or little time on each pattern as they need. The users are not informed of the specific mapping and are encouraged to build any correspondences that they naturally percieve, regardless of whether they are actually relevant or not. This ensures that the test truly measures the intuitiveness of the mapping and not how well a user can pinpoint symmetries only after being instructed. It is hoped that the generated song will be naturally mapped to the visual symmetries.

For the two base mappings and the motif mapping, the following test format is used:

- For each question a wallpaper group is chosen as the correct answer.
- The corresponding generated song is played according to the mapping under test.
- The user is presented with a binary choice between two wallpaper patterns, one of which comes from the correct wallpaper group and one does not.
- Once the user chooses an answer, the choice and the time taken on the question are anonymously recorded and the above steps are repeated for twenty total questions.

The users are not informed of the fact that time taken on each question is recorded. An unlimited practice test is also available to the user as a way of practicing the test format. It is identical in configuration to the real listening test above but can be answered indefinitely (questions are continuously generated) and feedback is provided on what the correct answer is. The questions, answers, user choices, and time spent on each question are all still recorded. The users are not informed that these results are recorded. Twenty questions is a balance between a short, manageable test, and ensuring that each wallpaper group appears enough times in the test.

Chapter 3

Software Design

3.1 Approach

The software behind this project spans multiple scopes from computer vision functions to audio generation and processing. Cross-compatibility and active support are the main criteria for selecting libraries as all the source code needs to readily incorporate into real platforms. The target integration platform for future systems stemming from WallpaperSynth is Linux-based and has processing power comparable to a microcomputer such as a RaspberryPi. However, streamlined data collection is also a goal and therefore requires WallpaperSynth to be deployable as a web application.

Local Application	**Web Application**
Integrable	Well-designed UI
High performance	Streamlined for data collection
Complete features	Informative

Table 3.1: The distinguishing software design requirements for local vs. web ports

This dual requirement stemmed the following pick-and-roll type of software development: First, any prototyping is done locally using desktop frameworks and languages. Once a feature is ready to be integrated to the data collection stage, the functionality is ported over to a web framework and deployed for ease of data collection. Anyone with access to a browser and the link should be able to access WallpaperSynth and its discernment listening tests. After data is collected and the validity of the feature is verified, it is incorporated back into the local application and flushed out for completeness and robustness. This ensures that the local application, which is the most essential deliverable, stays ready for use as an integrable module. A major requirement for both the local and web application is that the music generation module must support the ability to output one continuously generated song while changing the characteristics and structure of the song in real-time.

3.2 Modules

The overall functionality of WallpaperSynth can be divided into the modules seen in Figure 3.1. The discernment test module is for mapping research and development purposes only and would not be included in an actual system integration.

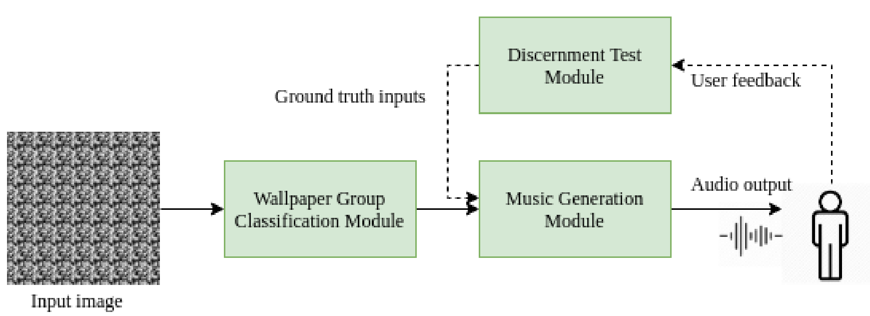


Figure 3.1: WallpaperSynth modules pipeline

3.2.1 Wallpaper Group Classifier Module

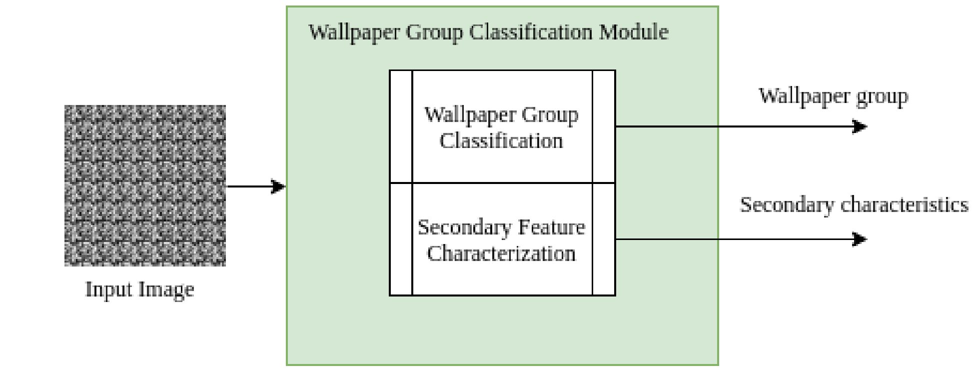


Figure 3.2: Wallpaper group classifier module pipeline

The purpose of this module is to extract and output visual characteristics from the input image, most importantly the wallpaper group of the dominant region of symmetry. The specific algorithms for symmetry and wallpaper group classification are interchangeable. Additional image processing extracts secondary characteristics from the input image. This entire set of features is passed along to the music generation module.

3.2.2 Music Generation Module

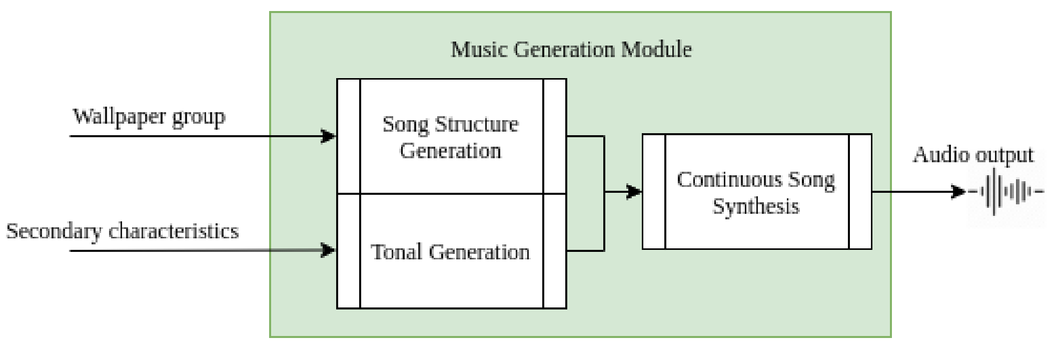


Figure 3.3: Music generation module pipeline

The main music generation module performs the pre-designed mapping from visual characteristics to music generation. Playback, song structure generation, and tonal control, are all responsibilities of this module. This module keeps the entire generated track on a looping timeline that abstracts away real time. The track consists of synthesized instruments playing sequences of notes and asynchronous programming is leveraged to change both the sequence and tone of the synthesized instruments depending on the visual characteristics output from the Wallpaper Group Classifier Module.

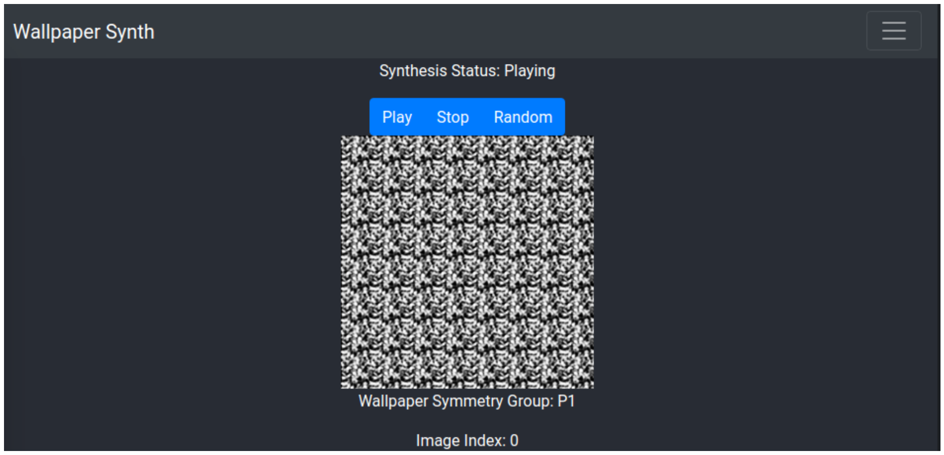


Figure 3.4: Screenshot of the WallpaperSynth web player user interface.

3.2.3 Discernment Test Module

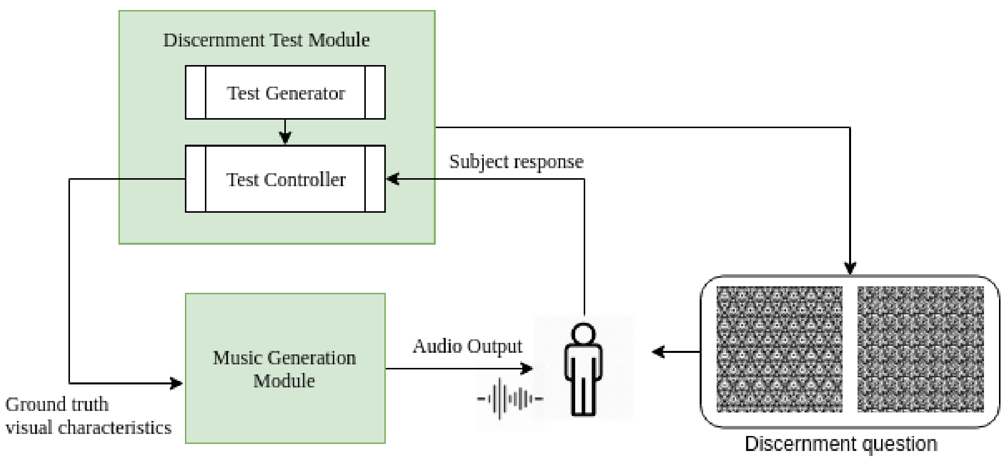


Figure 3.5: Discernment test module pipeline

This module generates and controls the listening tests for evaluating mapping effectiveness. It wraps around the music generation module by feeding it ground truth values for visual characteristics while simultaneously displaying the potential visual inputs that make up the current multiple-choice question. It handles user choice selection and persists the raw test results at test completion.

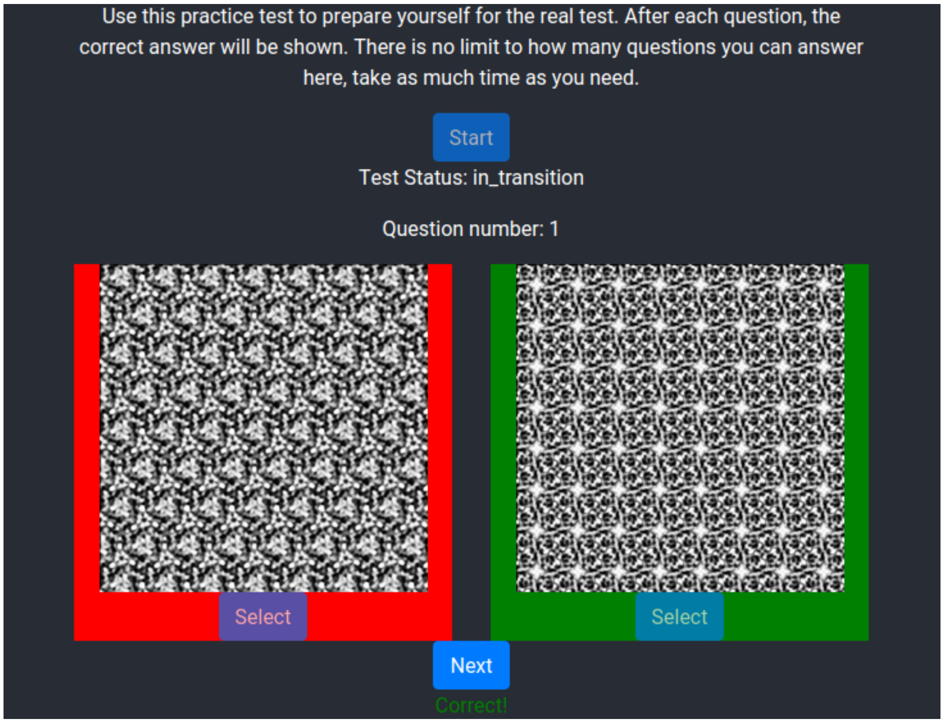


Figure 3.6: Screenshot of the WallpaperSynth web practice test user interface. Feedback is given for the correct answer.

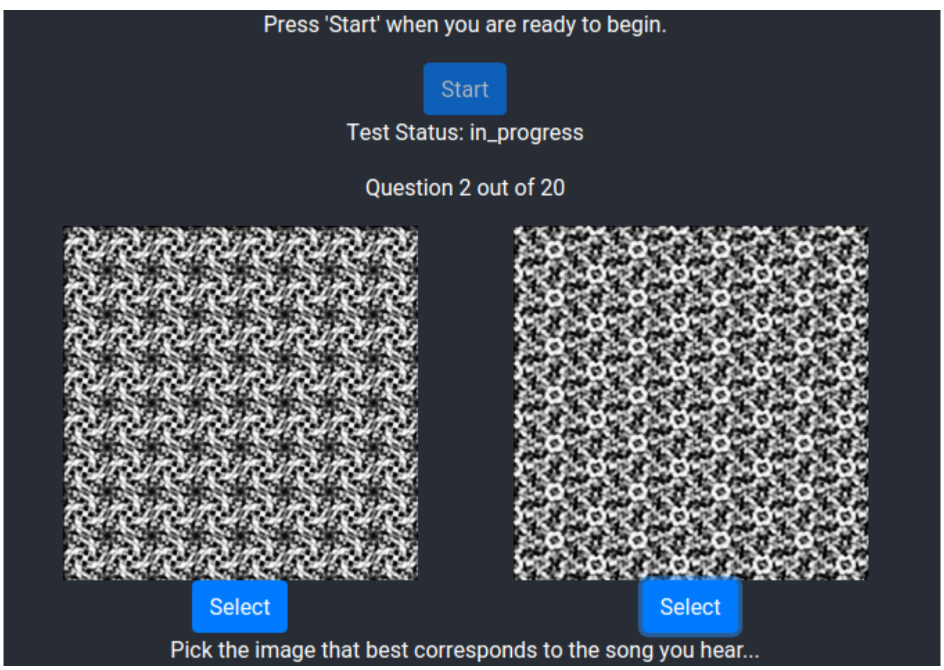


Figure 3.7: Screenshot of the WallpaperSynth web discernment test user interface.

3.3 Local Application

The local version of the WallpaperSynth application is a inter-dependent set of souce code, frameworks, and local servers that implement the functionality of the wallpaper group classifier, music generation, and discernment test modules.

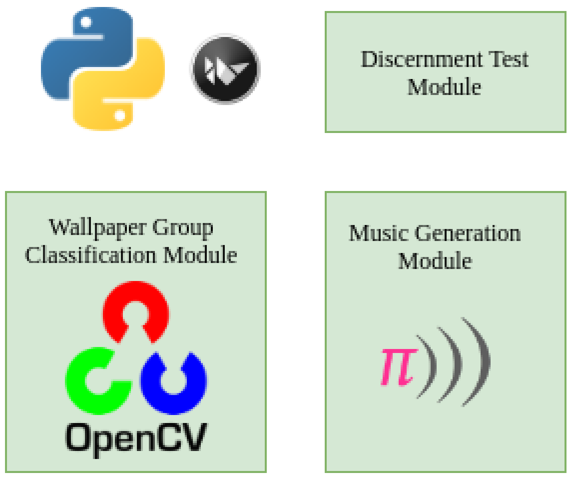


Figure 3.8: Local application software stack

3.3.1 Main Control Module (Python + Kivy)

Python was chosen as the language for the main control module for its versatility and abundant compatibility with other frameworks. Kivy, a user-interface library for Python, is used to create a GUI that presents the operation of WallpaperSynth and allows a user to control the local application. The overall pipeline of the application, along with the discernment listener tests, is implemented by this control module.

3.3.2 Music Coding Environment (Sonic-Pi)

In order to avoid reinventing the wheel, a music coding environment is needed to programmatically express the generation of digital music. Instead of working at the audio buffer and waveform level, a music coding environment works at the note, rhythm, and instrument level. Sonic-Pi is a live music coding environment built using the Ruby language and exposes a OSC server which allows easy connection from the control module via OSC client. Its live nature and cue/sync support means it has strong support for continuous song generation with real-time on-rhythm changes to structure or tonal characteristics. Under the hood, it is built on top of the SuperCollider audio synthesis engine.

3.3.3 Image Processing (OpenCV)

The pipeline starts with the classification and characterization of the input image. This is done by the OpenCV library which provides efficient and powerful implementations of common image processing functions. Most importantly, the music generation module uses OpenCV to classify the image to a wallpaper group, but it also extracts and quantifies secondary visual characteristics. This library is accessed through bindings in the main Python control module.

3.4 Web Application

The web version of the WallpaperSynth application is a full-stack web application, and still implements the functionality of the wallpaper group classifier, music generation, and discernment test modules. The frontend, which runs entirely in-browser, serves the same purpose of the desktop application while the backend provides a create, read, update, destroy (CRUD) API for discernment listening test results. The entire stack is hosted on AWS and accessible at www.wallpapersynth.com.

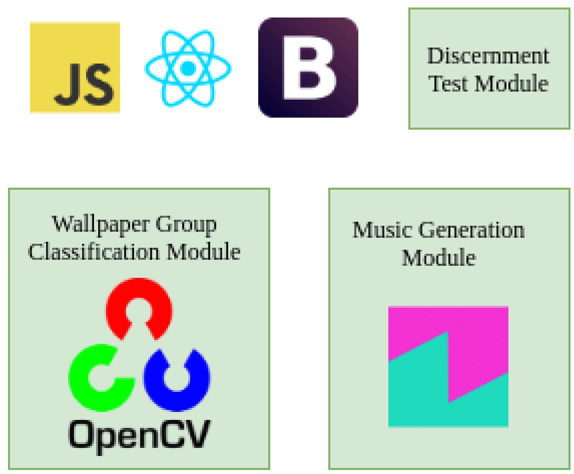


Figure 3.9: Web application software stack

3.4.1 Main Control Module (Javascript + React + Bootstrap)

With Javascript being the dominant in-browser language at the moment, it is used for writing the main control module of the application. React is a component-based frontend framework for the web and its one-way data binding philosophy works well with WallpaperSynth. Through React, the user interface is able to achieve the same level of responsiveness as the native desktop application. The Bootstrap CSS framework ensures that the user interface is responsive and dynamic on both desktop and mobile browsers. The entire frontend app is built as a static site and deployed using an AWS S3 Bucket.

3.4.2 Mosic Coding Environment (Tone.js)

Since the music generation needs to occur entirely in the browser, Tone.js is a great fit as it provides the same depth of high-level music synthesis as the Sonic-Pi library does. Tone.js is a Javascript library built on top of the Web Audio API, a platform for audio control included in nearly every single modern browser. Through Tone.js, which schedules sound events on a shared timeline called the Transport, a continuous song is generated with capabilities for real-time seamless changes to virtually any component of the song. This is integrated into the frontend app as an npm module.

3.4.3 Image Processing (OpenCV)

The OpenCV library is again utilized for the image processing components of the pipeline. OpenCV provides a subset of its full feature set through Javascript bindings. This allows for efficient wallpaper group classification and secondary feature characterization directly in the browser. This is integrated into the frontend app as an npm module.

3.4.4 Song Test Api (Ruby on Rails)

The discernment listening test results are stored through a RESTful API created using Ruby on Rails. The API stores the created tests in a PostgreSQL database. Test results can be retrieved individually, all at once, or grouped by test type, which enables continual monitoring and analysis of the data collection results. This backend application is deployed on an AWS EC2 instance and is served up by a Phusion Passenger + NGINX web server. The backend database is deployed on an RDS instance using the PostgreSQL 10.6 engine.

Chapter 4

Results

4.1 Variability and External Factors

The discernment listening tests provide insight on how well the correspondences are formed in a user's mind, in terms of both accuracy and time needed for recognition. Due to the binary nature of the test format, the hard baseline is 50% accuracy, which represents the expected outcome over multiple tests if the users simply guessed between the two choices. It is harder to determine a hard baseline for time taken on each question, but comparative analysis can reveal the relative time spent on different wallpaper groups on average.

Due to the crowd-sourcing nature of the test distribution, there isn't strict control on testing environment. This was a deliberate trade-off decision to increase response and acquire a larger, more representative sample of test results. The fact that the test is distributed as a web application that is accessible from both desktop and mobile browsers increases a large degree of variability that influences both accuracy and time spent on each question. The difference between using mouse and touchscreen may be minor but definitely present. There is no control on the listening environment either. Although headphone use is recommended, there is nothing blocking non-headphone users from completing the test. Without headphones, different speakers from different devices have varying sound signatures, and could affect the way the generated song is perceived.

There are many factors that may specifically affect accuracy on a per question and per user basis, many of which are human-related. For example, a question may present two symmetry patterns that are from different groups but look very similar due to the textures. If the correspondence formed in the user's mind leans more towards textures than symmetry, then the user may have to resort to a random guess. Additionally, a user may learn more about the correspondence and gain confidence while doing the practice test and therefore answer a higher proportion correct as the test progresses. The external factors that influence accuracy are not limited to these examples. However, given a large enough set of people taking the tests, the average accuracy should be representative of the true level of effectiveness.

Many factors also affect the time spent on each question, and again they are human related. This is especially variable because the users don't know their time is being recorded as a metric. As users answer more questions on an exam, they may feel more comfortable with the format and answer faster. Once the they know the layout of the buttons and know where to look, they may spend less time on each question. They may speed up towards the end simply because they want to finish the test. The user may need to step away from the assessment in the middle of a question and come back, leading to an outlier. Again, with a large enough set of people taking the test, the normalized average elapsed times will provide meaninful comparison.

4.2 Testing Demographic

The web app was sent to friends, family, and Laboratory for Perception, Action, and Cognition members. Although no demographic information was explicitly collected, the set of people who the test was sent out to have varying ranges of musical ability, a good representation of genders, and a wide variety of backgrounds. Most of the participants were college students.

4.3 Overview

Accuracy for a given test is as follows:

$$accuracy = \frac{\text{number of correct answers}}{\text{number of total answers}} \tag{4.1}$$

Accuracy provides a basic measure of mapping effectiveness through level of discernment.

Test Type	Response Count	Average Accuracy	Questions per Test
base-mapping-v1	5	0.48	10
base-mapping-v2	10	0.50	20
base-mapping-v2-practice	8	0.53	Variable
motif-mapping-v1	17	0.61	20
motif-mapping-v1-practice	21	0.62	Variable

Table 4.1: Overview of accuracy and number of responses by test type.

Although the base mappings are at the hard baseline, the motif mapping shows a roughly 10% accuracy improvement over random guessing. On average the base mappings essentially provide no notable correspondence, while the motif mapping does offer some level of intuitive correspondence.

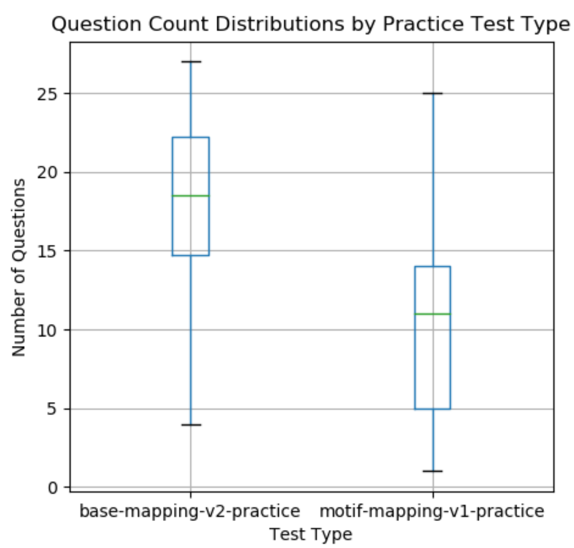


Figure 4.1: The distribution of question counts for tests in the two practice test mappings. The number of questions to answer during a practice test session is completely up to the user.

On average, users spent a higher number of questions on the practice test for the base mapping over the motif mapping. This suggests it takes longer for the users to feel confident (or as confident as possible) about the correspondences they are forming for the base mapping compared to the motif mapping. No question count distribution is shown for the actual tests because they have a fixed number of questions.

Test Type	P1	P2	PM	PG	CM	PMM	PMG
base-mapping-v1	5	2	5	10	12	6	10
base-mapping-v2	30	34	33	27	25	24	27
base-mapping-v2-practice	25	26	24	21	16	19	13

Table 4.2: The total number of times each wallpaper group appeared as the correct answer across all base mapping tests

Test Type	P1	P2	P3	P4	P6
motif-mapping-v1	51	73	73	73	70
motif-mapping-v1-practice	38	38	45	56	38

Table 4.3: The total number of times each wallpaper group appeared as the correct answer across all motif mapping tests

4.4 Accuracy by Test Type

Comparison of accuracy distributions reveals a significant amount of information about mapping effectiveness. Base mapping v1 only has five samples and therefore is not as representative of the population as the other mappings.

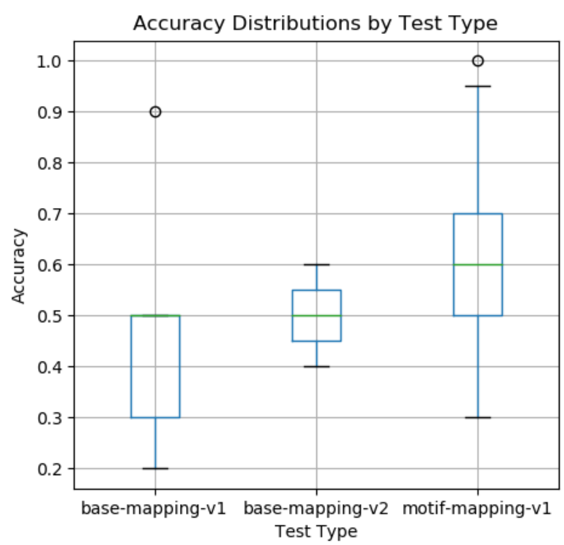


Figure 4.2: The distributions of per-test accuracy by test type.

A narrow spread, as is the case for base mapping v2, means that the sampled users performed similarly to each other. The mapping likely had similar effectiveness on all users who took the base mapping v2 version of the discernment test. The fact that the accuracies are centered around 0.50 suggests users didn't build a much better or worse correspondence than random guessing. Motif mapping v1 has a relatively larger spread which means that the mapping had a wider range of effectiveness on users. The first quartile is located at 0.50 and the median is at 0.60, which is indicative of stronger discriminatory capabilities for some users and essentially random guessing for others. With the third quartile at 0.70 and even a perfect score outlier, a handful of users developed a very strong discernment from motif mapping v1.

4.4.1 Base Mapping v2 vs. Motif Mapping v1 T-test

A two-tailed t-test for populations with different variances was carried out between the test samples from base mapping v2 and motif mapping v1. The null hypothesis is that users achieve the same mean accuracy on tests from these two mappings and the alternate hypothesis is they do not. A t value of -1.92 and a p value of 0.068 were acquired. Using a valid significance value of 0.10, this is enough to reject the null hypothesis. However, using a significance value of 0.05, the null hypothesis would fail to be rejected. This mounts enough evidence to believe that users are able to, on average, achieve a better identification accuracy through the motif mapping than the base mapping. The motif mapping is an experimental demonstration showing users are capable of build some level of symmetry correspondence between visual and auditory perception in an undirected manner.

4.5 Accuracy by Wallpaper Group Within Test Type

With a large enough sample size of test responses, each specific wallpaper group mapping within the whole mapping can be evaluated independently. These relative accuracies reveal which wallpaper groups were easier or harder to discern for the user, and can pinpoint the specific sub-mappings that should be targeted for improvement. The accuracy for a specific group for a single test is calculated by dividing the number of times that group both appeared as the correct answer and was answered correctly by the number of times the group appeared as the correct answer. The overall average accuracies by wallpaper group are found in Tables 4.4 and 4.5. The distributions of single test accuracies by wallpaper group are found in Figures 4.3 and 4.4

4.5.1 Base Mappings

Test Type	P1	P2	PM	PG	CM	PMM	PMG
base-mapping-v1	0.60	0.00	0.60	0.60	0.17	0.50	0.70
base-mapping-v2	0.47	0.44	0.52	0.59	0.40	0.46	0.63
base-mapping-v2-practice	0.64	0.35	0.46	0.71	0.44	0.47	0.77

Table 4.4: The average accuracies by wallpaper group for the base mappings.

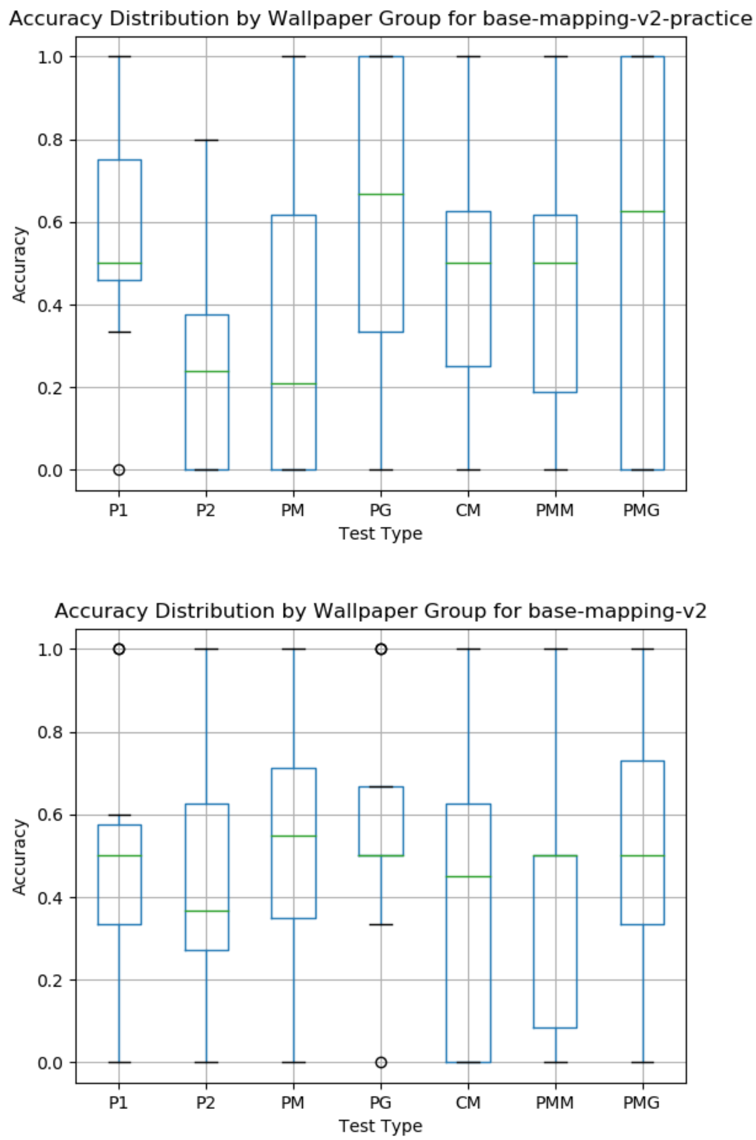


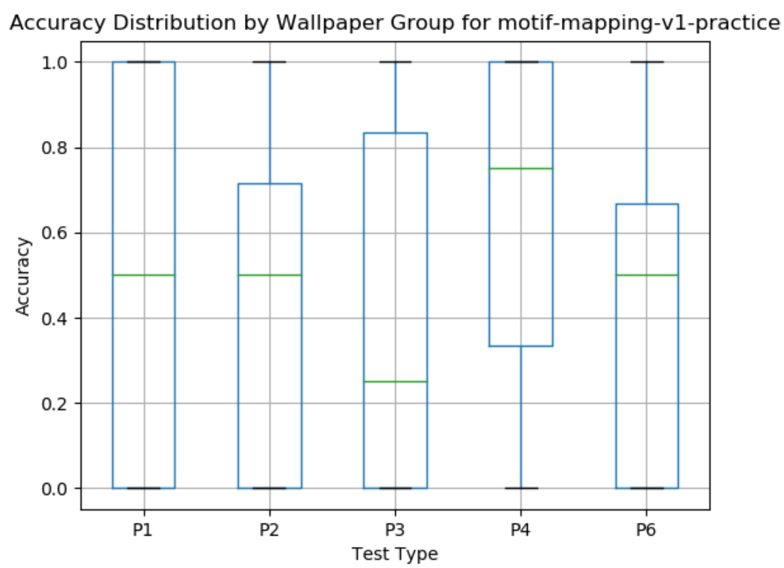
Figure 4.3: The distributions of single-test accuracies by wallpaper group for base mapping v2 (right) and the corresponding practice test (left).

Although the single-test accuracy distributions have too much variability due to the fact that each group only appears in each test as the right answer on average two to three times, the overall average accuracies by group definitely reveal that certain groups were easier to identify than others. For base mapping v2, the average accuracies follow similar trends for the real test and practice test. The PMG and PG groups are identified more accurately than other groups, for both real and practice tests. For most groups, the average accuracy is higher for the practice test, likely due to the fact that the user recieves feedback on the correct choice immediately after each question. The user could potentially use trial and error techniques to perform better on the practice test. This is a highly interesting result especially since each group is represented by a different pitch with the same tonal qualities and rhythm and only a minute percentage of people have perfect pitch.

4.5.2 Motif Mapping

Test Type	P1	P2	P3	P4	P6
motif-mapping-v1	0.55	0.62	0.58	0.68	0.60
motif-mapping-v1-practice	0.55	0.68	0.58	0.68	0.61

Table 4.5: The average accuracies by wallpaper group for the motif mapping.



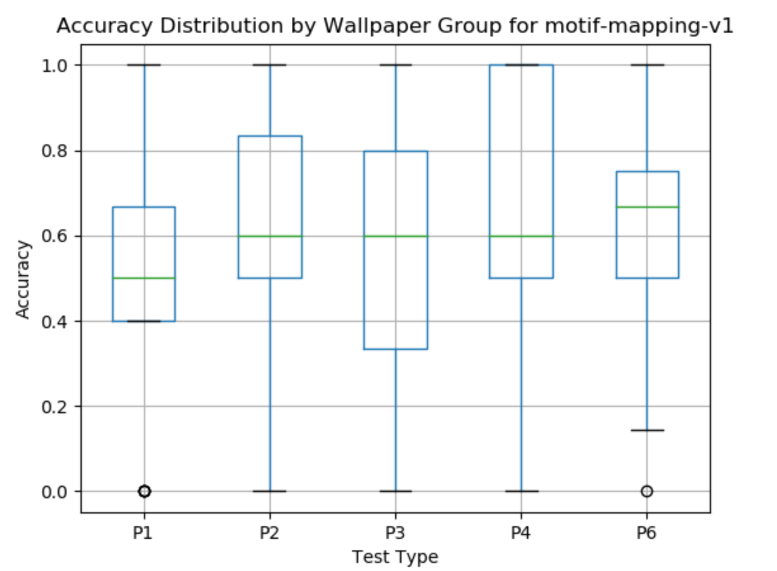


Figure 4.4: The distributions of single-test accuracies by wallpaper group for motif mapping v1 (right) and the corresponding practice test (left).

The motif mapping has nearly congruent average accuracies by group between the real test and the practice test. This indicates that the correctness feedback after each question on the practice test has relatively low effect on identification accuracy. The only group that had a major difference was P2, where the practice test had a higher average accuracy. This could imply that P2 is identified more accurately when prior correct examples of P2 patterns are seen within the same test. Based on average accuracies, the P4 group had the most discriminative mapping. It could also simply imply that the P4 wallpaper group has a more distinct symmetry than the other groups.

The single-test accuracy distributions are more meaningful for the real test than for the practice test. The practice test has a user-determined length and could result in certain groups not not even showing up at all as the correct answer on short tests. The distributions reflect the overall accuracies well for the real test with the exception of the P4 and P6 groups. These discrepancies likely stem from low group representation in specific tests which leads to random variance.

4.6 Duration By Test Type

Test Type	Average Test Duration (Seconds)
base-mapping-v1	65.96
base-mapping-v2	71.75
motif-mapping-v1	135.17

Table 4.6: The average test duration for the test types with a fixed number of questions.

The only valid comparison to be made is between the base mapping v2 and motif mapping v1 average durations, as they have the same number of questions per test (20). The motif mapping tests took much longer likely due to the fact that the distinguishing melodical element lasts for an entire measure, compared to an 8^{th} note for the base mapping.

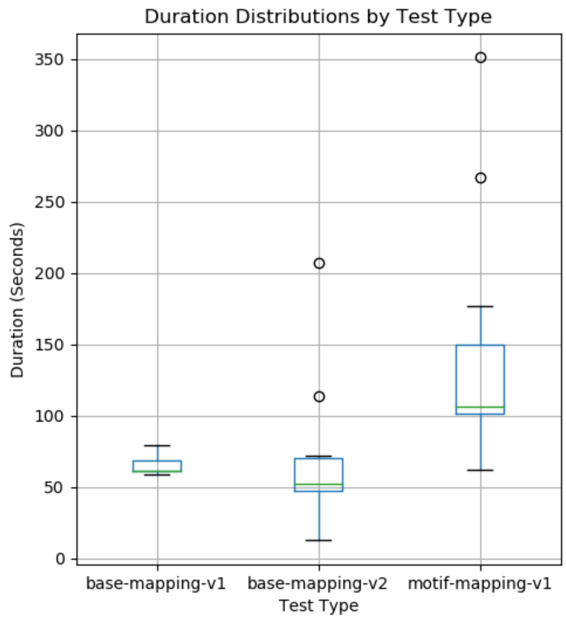


Figure 4.5: The distributions of per-test duration by test type.

However, the box plots in Figure 4.5 reveal that the average duration is being pulled up by a number of outliers. The same phenomenon is observed when comparing medians so the above conclusion still holds.

4.7 Elapsed Times By Test Type

As seen in Table 4.7, the average elapsed times per question are much higher for the two practice tests compared to the real tests, likely due to users not strictly focused on the questions the entire time. Once it came time for the real test, they likely focused better. After all, they don't know that elapsed times are being recorded, which was an intentional trade-off decision. Again, the average elapsed time for the real motif mapping v1 test is higher than for the real base mapping v2 test because the distinguishing melodical element lasts longer. It takes longer for a user to hear the entire motif and then make a decision.

Test Type	Average Elapsed Time Per Question (Seconds)
base-mapping-v1	6.60
base-mapping-v2	3.59
base-mapping-v2-practice	42.59
motif-mapping-v1	6.76
motif-mapping-v1-practice	62.70

Table 4.7: The average time spent on a question (in seconds) by test type.

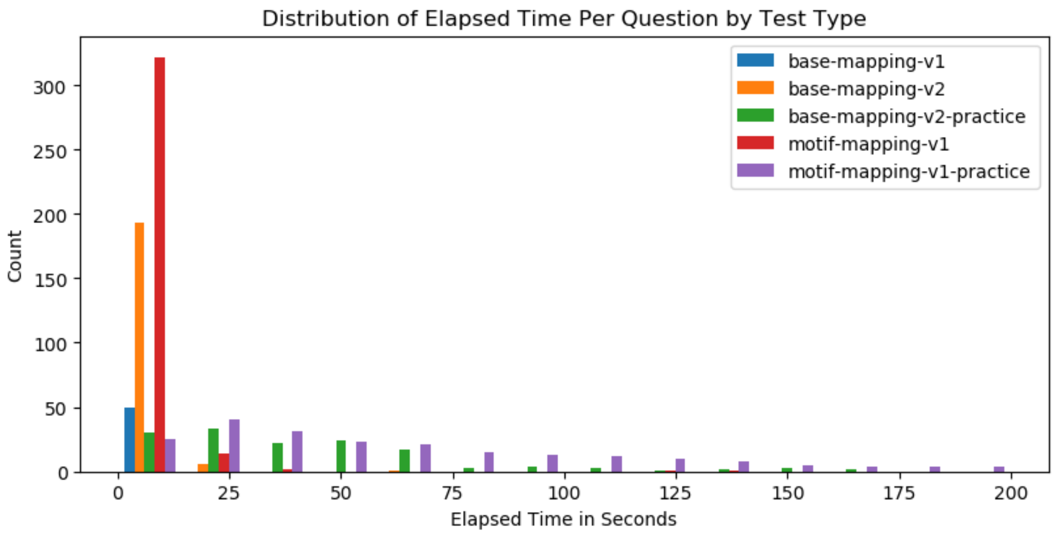


Figure 4.6: The distributions of per-question elapsed time by test type visualized as a joint histogram. The outliers for the practice tests have skewed the distribution

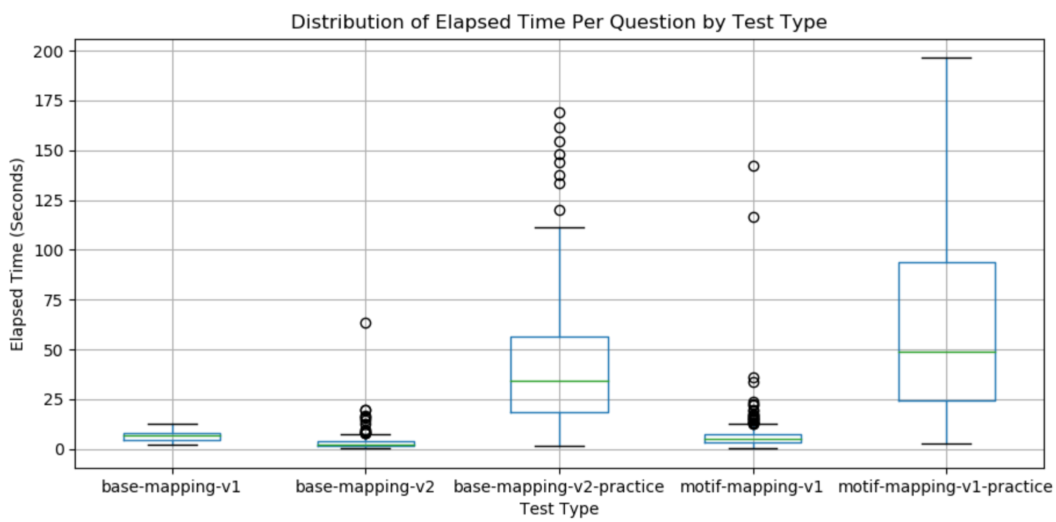


Figure 4.7: The distributions of per-question elapsed time by test type.

Although some outliers still exist, there is much smaller variance in elapsed time for the real tests compared to the practice tests. This suggests that users had slightly differing attitudes between the two categories and truly treated the practice test as a unpressured opportunity to practice.

4.8 Elapsed Times by Wallpaper Group Within Test Type

Comparing the effectiveness of different wallpaper groups within a mapping is done by looking at the time spent on a per-question basis and the aggregate statistics surrounding those elapsed times. Though the raw elapsed times can provide some insight (Table 4.8 and 4.10), it is more useful to compare elapsed times that have been normalized on a per-test basis (Table 4.9 and 4.11). This is so individual tendencies are taken into account and a valid comparison can be made between wallpaper groups. The values are normalized so that all elapsed times for a single test add up to 1.

For base mapping v2 and motif mapping v1, spending equal amounts of time on each question would result in a normalized average of 0.05 since there are twenty questions per test. For base mapping v1, spending equal amounts of time on each question would result in a normalized average of 0.10 since there are ten questions per test. For the real tests of the two base mappings, there appears to be no significant difference in average elapsed times. There are a number of outliers for base mapping v2, and may represent times where a user felt stumped. The CM, PMM, and PMG groups all had a couple of outliers, suggesting that when offered as the correct answer, songs generated from these groups may yield a more difficult correspondence.

4.8.1 Base Mappings

Test Type	P1	P2	PM	PG	CM	PMM	PMG
base-mapping-v1	6.36	6.31	4.47	7.04	7.48	7.21	5.96
base-mapping-v2	5.67	3.07	2.70	3.29	3.61	3.41	3.45
base-mapping-v2-practice	43.05	47.04	34.86	44.67	49.09	36.39	44.77

Table 4.8: The average per-question elapsed times (seconds) by wallpaper group for the base mappings.

Test Type	P1	P2	PM	PG	CM	PMM	PMG
base-mapping-v1	0.092	0.106	0.068	0.107	0.115	0.114	0.086
base-mapping-v2	0.053	0.047	0.044	0.041	0.048	0.055	0.064

Table 4.9: The average per-question elapsed times (seconds) by wallpaper group for the base mappings, normalized on a per-test basis.

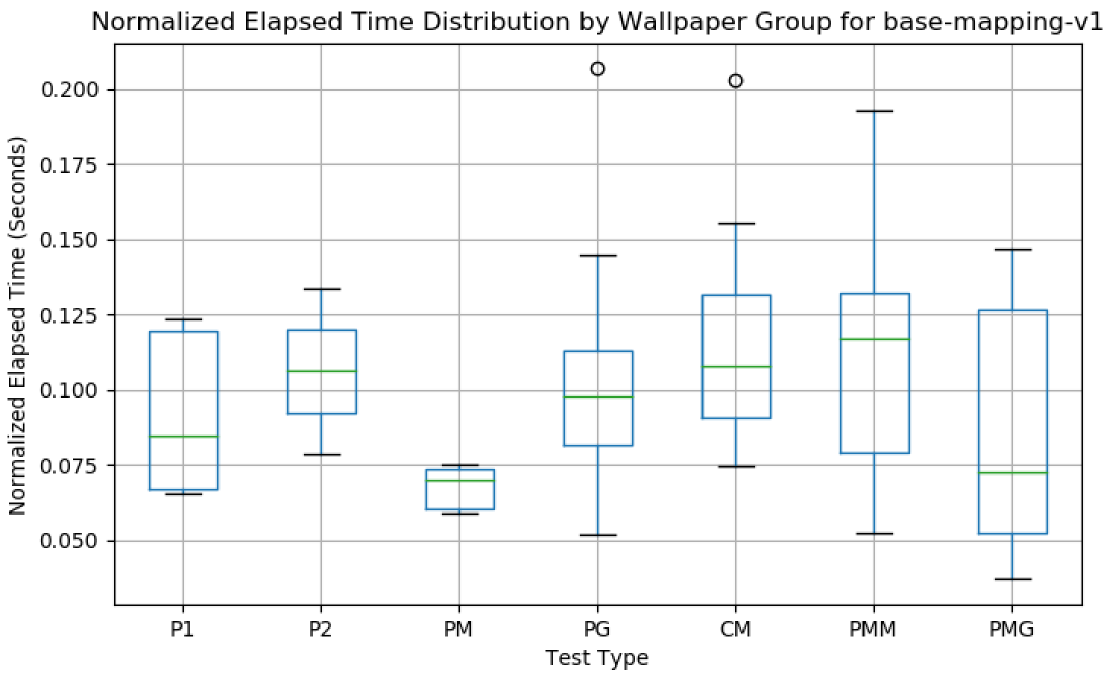


Figure 4.8: The distribution of normalized per-question elapsed times for base mapping v1.

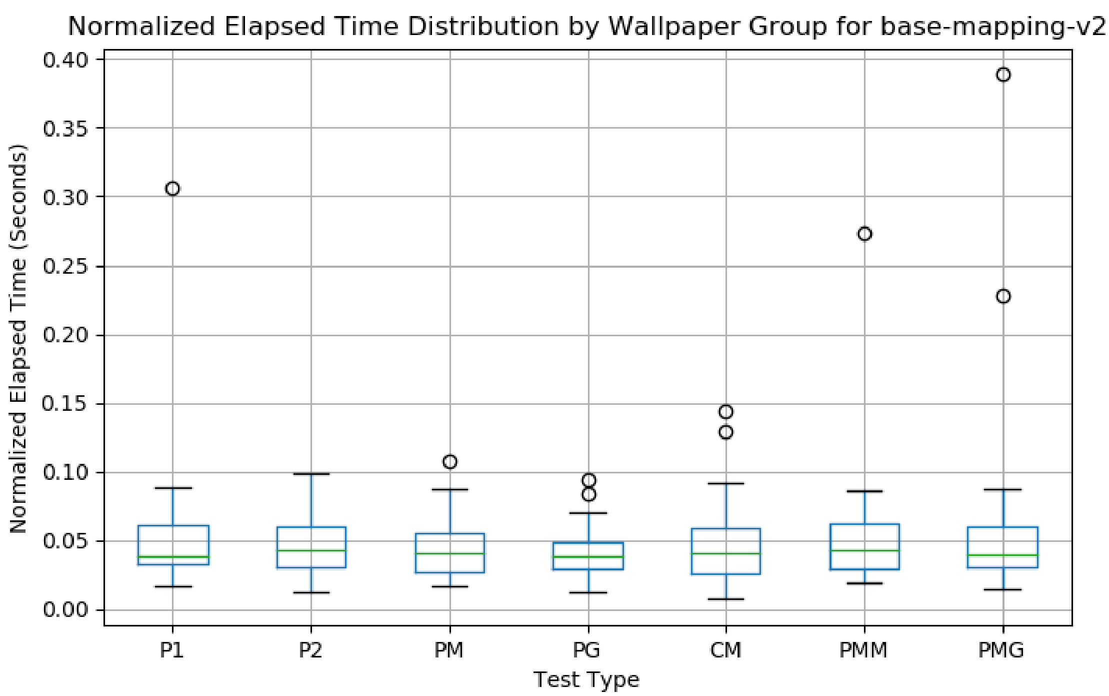


Figure 4.9: The distribution of normalized per-question elapsed times for base mapping v2.

4.8.2 Motif Mapping

There appears to be no significant variation in average elapsed times for the motif mapping either, although the P4 groups and P6 groups had slightly shorter times than the expected baseline. This may suggest that the P4 and P6 groups lead to slightly faster and more confident identification than the others. Subjectively, it seems as though the patterns from these two groups are more distinct than patterns from the other three, due to the way the 4- and 6-fold rotation centers stand out. This could also suggest the generated melody motifs for P4 and P6 are more distinct than the others.

Test Type	**P1**	**P2**	**P3**	**P4**	**P6**
motif-mapping-v1	8.21	6.52	6.65	7.02	5.79
motif-mapping-v1-practice	62.29	62.49	68.52	53.73	69.64

Table 4.10: The average per-question elapsed times (seconds) by wallpaper group for the motif mapping.

Test Type	**P1**	**P2**	**P3**	**P4**	**P6**
motif-mapping-v1	0.056	0.050	0.052	0.048	0.045

Table 4.11: The average per-question elapsed times (seconds) by wallpaper group for the motif mapping, normalized on a per-test basis.

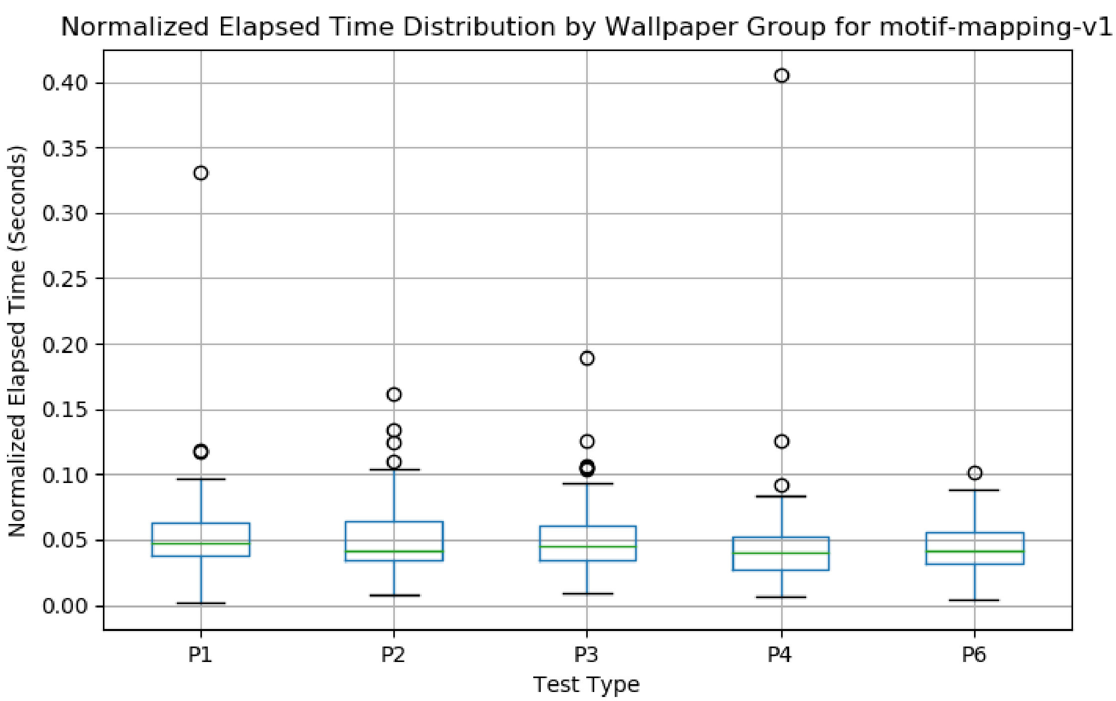


Figure 4.10: The distribution of normalized per-question elapsed times for motif mapping v1.

Chapter 5

Conclusion

5.1 Outcomes

This thesis has resulted in the following set of outcomes:

- A prototype system to continuously generate music in response to the symmetry and secondary visual characteristics present in an input image.
- An efficient, crowd-sourced means to test the absolute and comparative discernment power of various visual to auditory mappings.
- A specific melody motif-based mapping that offers statistically significant improvement over a hard baseline discernment level, which suggests some level of natural linkage between visual and auditory symmetry perception.

Due to the lack of regulation for the testing environment and a number of factors that are intrinsic and extrinsic to the design of the tests, the results derived from this thesis should be treated as what they are: an exploratory approach that uses experimental means to observe the effects of a audio-visual symmetry mapping.

5.2 Future Work

5.2.1 Improved Mappings

The motif mapping presented in this work is extremely rudimentary compared to the limitless possibilities afforded by the world of music. With modern synthesizers and effects, a much more sophisticated collection of sounds and musical elements can be produced. These details, though subtle, are similar in nature to the level of detail and style perception humans have with vision. Future work would focus on increasing the complexity and depth of mapping coherence, without sacrificing intuitiveness. Ultimately, the apex goal would be to find a mapping technique that enables the auditory perception of visual elements at native levels of detail.

5.2.2 Rethinking of Approach

The discernment listening tests presented by this thesis may not be the ideal way to test the effectiveness of the mappings. It is not grounded in any neurological basis, and instead treats human perception as a black box. Given the prior work on the neural basis of visual symmetries [2] and visual cortex representation of wallpaper symmetries [8], it would be interesting to use a more direct physiological approach to analyze actual neurological linkage for symmetry perception instead of relying on speculation.

5.2.3 Machine Learning

Machine learning can be employed to help build mappings and test the effectiveness of mappings. Although limited prior work exists on this technique, GANs could be trained to generate music automatically from a set of desired parameters. A neural network that mirrors human hearing (Gammatone filterbank + LSTM Autoencoder) [9] could help determine the effectiveness of new mappings in an automoted way. Finally, evolutionary algorithms can be used to automate the process of synthesizing and evaluating new mappings. This technique does not have to detract from the benefits provided by the manual creation of mappings, but instead could embed rules that ensure the mappings are musical and human-friendly.

5.3 Conclusion

There is abundant value to be derived in researching groups of human perceptive senses in conjunction, not just independently. While investigating the differences between perceptive senses can bring more understanding about the specific potential of each sense, investigating the similarities and connectivity can reveal more about human perception as a whole. Visual perception is extremely detailed and and plays a disproportionately large role in object recognition and classification. The human brain's ability to pre-attentively recognize symmetry highlights the importance of information abstraction and the hierarchy of information retrieval from perception. By relating high-level visual symmetries to high-level musical symmetries, the way the brain processes general symmetry was explored experimentally.

A system was devised to generate a continuous musical track in response to the symmetries and secondary visual characteristics present in an input image. A melody motif-based mapping was invented for this system that provides correspondence between wallpaper symmetry groups and the generated song. This mapping demonstrated an effectiveness level above baseline levels in human discernment tests, and proves the possibility of mapping between visual and auditory perception. The use of audio and music as a means of letting one perceive symmetries and textures in a general manner has valuable research and practical applications that could make the world a more accessible and dynamic place for all humans.

Bibliography

[1] H. Weyl, *Symmetry*. Princeton University Press, 1956.

[2] M. Bertamini, J. Silvanto, A. M. Norcia, A. D. Makin, and J. Wagemans, “The neural basis of visual symmetry and its role in mid- and high-level visual processing,” *Annals of the New York Academy of Sciences*, vol. 1426, no. 1, p. 111126, 2018.

[3] Y. Liu, H. Hel-Or, C. S. Kaplan, and L. Van Gool, “Computational symmetry in computer vision and computer graphics,” *Foundations and Trends in Computer Graphics and Vision*, vol. 5, pp. 1–195, 2010.

[4] “Wallpaper groups.” [Online]. Available: http://mathworld.wolfram.com/WallpaperGroups.html

[5] “Sound and music.” [Online]. Available: http://www.howmusicworks.org/104/Sound-and-Music/Musical-Instrument-Tone

[6] “Symmetry and centricity open music theory.” [Online]. Available: http://openmusictheory.com/symmetryAndCentricity.html

[7] A. Kak, “Measuring texture and color in images.” [Online]. Available: https://engineering.purdue.edu/kak/Tutorials/TextureAndColor.pdf

[8] K. Peter, A. Clarke, Y. , Y. Liu, and A. Norcia, “Representation of maximally regular textures in human visual cortex (the journal of neuroscience),” *The Journal of Neuroscience*, vol. 36, pp. 714–729, 01 2016.

[9] D. Rothmann and D. Rothmann, “Human-like machine hearing with ai (3/3),” Feb 2019. [Online]. Available: https://towardsdatascience.com/human-like-machine-hearing-with-ai-3-3-fd6238426416

Andy Luo Academic Vitae

andy.yifan.luo@gmail.com | 814·441·7638

EDUCATION

PENN STATE UNIVERSITY
BS Computer Science
BS Electrical Engineering
Aug. 2015 - Pres. | State College, PA
Schreyer Honors College
Leonhard Engineering Scholars Program
Presidential Leadership Academy
Dean's List (All Semesters)

COURSEWORK

ELECTRICAL ENGINEERING

Electronic Circuit Design I, II & III
Nanoelectronics • Electromagnetics
Cont. Time Linear Systems
Computer Vision I & 2 • Design Tools
Digital Image Processing

COMPUTER SCIENCE

Digital Design • Computer Organization
OOP with Web Applications
Systems Programming • Discrete Math
Data Struct. & Algs. • Operating Systems
Theory of Computation • Security
Software Design Methods
Pattern Recognition

SKILLS

OVERVIEW

Analog Circuitry • Microelectr. & Optics
PCB design • Control Systems
Embedded Systems • Application Dev.
Web Development • Machine Learning

COMPUTER LANGUAGES

Java • C++/C • Python • Ruby • Scala
SQL • HTML • CSS • Javascript
Assembly • Verilog • SOAR • $\LaTeX$

FRAMEWORKS & LIBRARIES

Ruby on Rails • Bootstrap • React
Android • Apache Spark • Apache Kafka
MySQL • PostgreSQL • PostGIS

TECHNICAL APPLICATIONS

Labview • Altium • Xilinx Vivado
ARM Cortex Toolchains • Simetrix
Unity • Unreal Engine 4

CERTIFICATIONS

Certified Labview Associate Developer

HUMAN LANGUAGES

English • Mandarin

EXPERIENCE

CAPITAL ONE | Software Engineering Intern

Jun – Aug 2018 | McLean, VA

- Provisioned the engineering and operational harness necessary for deploying a cloud-based real-time malware detection pipeline
- Benchmarked the performance of Apache Flink vs. Apache Spark for domain-specific data and operators
- Developed a tool for generating synthetic network log messages, up to 1 million/sec, used for benchmarking and validation of malware detection model

SIEMENS EM | Electronic Hardware R&D Intern

May – Aug 2017 | Norcross, GA

- Conceptualized, designed, and fabricated RFID reader capabilities for smart grid electric vehicle chargers
- Improved hardware features on a prototype electric vehicle simulator to drastically reduce test durations and introduce automated cyclic testing
- Prototyped the full-stack cloud infrastructure to implement a distributed peak shaving service, featuring low-latency data stream processing and C2D/D2C communications

CROWDSHOUT | Founder & Developer

Dec 2016 - Present | State College, PA

- Devised MVP and business model for geolocation-based social media platform that aggregates spatially and temporally relevant content for users
- Led team of 2-3 developers in realizing content aggregation algorithms, a non-linear content feed, and a cloud-hosted RoR back-end framework

RESEARCH

LAB FOR PERCEPTION, ACTION, AND COGNITION | Researcher

Sep 2018 – Present | State College, PA

- Created mappings for 2D wallpaper symmetry groups from visual to auditory domain through musical synthesis
- Collected, cleaned, and validated Taiji motion capture data through Vicon

NAVY APPLIED RESEARCH LAB | Researcher

May 2016 – Dec 2017 | State College, PA

- Coded control system elements, realized hardware implementations, and performed weekly in-water tests for a multi-agent autonomous vehicle software architecture with natural language capabilities
- Developed database and cache-based container for storage, processing, and retrieval of shared tactical spatial data

LEADERSHIP & ORGANIZATIONS

2015-2018	President	Audio Engineering Society Student Chapter
2016	GNC Engineer	Penn State Lunar Lion
2015-Present	Member	IEEE Student Chapter

PROJECTS

2017	HackPSU: 1st place for ultrasonic walking stick
2017	HackPrinceton: Best use of Cloud Services for voice-controlled EE lab bench
2016	HackPSU: 1st place for network-enabled health wristband