

Cross-Domain Analogy: From Image to Music

Joana Borges Teixeira
joana.teixeira@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

May 2017

Abstract

Creativity and inspiration are two largely intertwined concepts. Whether we search for inspiration from inside us, the outside world or even from divine sources, it helps us when attempting to create something new or to solve a problem.

Not many attempts have been made of modeling and implementing an inspirational system. We called this a cross-domain analogy: starting with an artifact from one domain, its essence is transposed into a new domain.

In this Thesis, we propose a possible analogy between the visual and the musical domains. Furthermore, we have implemented a system that applies that analogy. It extracts a set of readily available features from an image given as input and translates them into two different musical artifacts. The first, called the raw version, represents the purest translation since little rules are imposed on it. The second, called the harmonized version, has some Rock constraints imposed on it, which results in a less pure translation. A third version is also generated, resulting from the execution of a Genetic Algorithm. The initial population is generated from the first two versions.

Two evaluations were performed at two different phases. The results led us to conclude that the system presented in this Thesis is able to generate artifacts that were regarded as music by the majority of the evaluators. Furthermore, the artifacts were liked by a great number of people. Finally, many people considered the artifacts reflected the images, although not so many as the ones who liked them.

Much work can still be done to further improve both the analogy and the system we presented. However, it is our belief that the proposed analogy provides an interesting and justifiable solution to translate images into music.

Keywords: Computational Creativity, Cross-Domain Analogy, Translation, Genetic Algorithm, Rock, Paintings

1. Introduction and Motivation

Humans are known to be inspired by their surroundings, which may result in the creation of something new. Although a relatively common process in our minds, only recently have researchers attempted to model and implement it in a program. Work in Computational Creativity focuses both on a theoretical perspective, such as the creation of models and frameworks to describe the creative process, or on a practical perspective, by developing systems that exhibit creative behavior. Both are important, and usually the latter is based on the results of the former. Among many possible sub-areas of Computational Creativity, cross-domain analogy, or inspiration, is still a mostly under-explored concept [1], even though we humans are usually inspired by our surroundings during our creative processes.

Mel Rhodes proposed the "four P's of creativity" – **Process**, **Product**, **Person**, and **Press** [7]. These terms can be used to loosely define creativity as "A process, executed by a person, pressed by

his/her environment, by which a product is generated". Therefore, inspiration can be considered the **press** aspect of creativity, since Rhodes defined it as how the environment affects the mind during the creative process.

Taking this into account, our goal was to create a system with potential to exhibit creative behavior, which uses visual input (such as an image, a photograph or a painting) as inspiration for generating a musical artifact. The development of our system required us to better understand each of the domains, individually. Several issues were analyzed: What features can we extract from an image, and how should they be interpreted? How can we attribute a specific emotional state to an image? What are the basic elements needed to create a music? What are the different features that a composer can use to create a specific emotional state in a music? Based on the answers to these questions, we created a bridge between the two domains.

Our main contribution is one possible translation

between features from the visual domain to the musical domain. By using these features, we were able to create artifacts that were regarded by evaluators as music.

Our work was divided into several phases. In an initial phase, we researched both domains. We studied color theory, and how each different color should be interpreted. Furthermore, we also learned about music theory, seeking to understand more about the components of a music. During this initial phase, we consulted musical experts, who gave us some guidance regarding the musical domain.

On a second phase, we analyzed both the features from the visual domain, and from the musical domain, and started relating them to one another. Every relation that was created was also implemented, so that we could always listen to the results and from them try to further improve the translation. At this point, we did not worry about the aesthetic value of the artifacts. In this phase, the raw and harmonized versions were born. An intermediate evaluation was performed at this point, to understand if we were advancing in the right direction.

Finally, having completed our cross-domain analogy and the implementation of two different versions, we attempted to improve the aesthetic value of the artifacts, while still preserving their novelty. To achieve this goal, a Genetic Algorithm was implemented, with a fitness function designed to prefer artifacts that followed certain guidelines generally known to be related to more aesthetically pleasing music.

It should be noted that we do not presume to label our final results as "music", since opinions on what actually music actually is vary throughout the musical community. For example, some consider that a musical artifact can only be called music if a human has worked on it. As such, our results may be considered to belonging to the synthesis domain – they are a set of sounds that can then be used by a human to create music. Throughout this Thesis, we will call our results as music, or musical artifact, but only in the most general sense: they are a series of sounds produced by different instruments that were brought together to create an artifact that may, or may not, be regarded as music, and that can be, eventually, worked on by a human.

In the next section, we provide some insight on previous work on inspiration and musical generation. Further on, we describe the features we have identified, both in the visual and the musical domains. Then, we explain how we have related these features to each other, creating our cross-domain analogy. In the following section, all the implementation details of the three versions are explained. Then, our final results are discussed. Finally, we conclude with some final notes on what we have

achieved, some possible applications for our system and future work to further improve it.

2. Related Work

The main focus of our work is the act of **Inspiration**, which happens frequently in our minds. However, neither Psychology [11] nor Computational Creativity [1] have dedicated much effort to the study of this topic. Thrash and Elliot [11] posit that there are two important objects during an act of inspiration: the trigger – the object that evokes inspiration – and the target – the object to which the motivation is directed. There are three possible sources of inspiration: Supernatural (divine inspiration); Intra-psychic (our own sub-conscious thoughts inspiring our consciousness); and Environmental (nature, other people or works of art).

2.1. Inspirational Systems

While Psychology attempts to understand and explain what inspiration is and how this process occurs, Computational Creativity attempts to simulate it with software. The goal is to create computational systems that receive a given input (from any domain – visual, musical, textual, etc) and generate a new artifact that tries to have the same mood, or characteristics (which may or may not be in the same domain).

Johnson and Ventura [2] created a system that composes musical motifs, which they describe as "the smallest structural unit possessing thematic identity". To create these motifs, non-musical media is used as inspiration, including non-musical sounds, such as a bird chirping or a running engine, and images. The created motifs may then be used by a human composer to create a full composition, for example. The input is analyzed, and several different candidate motifs are generated. To select one motif as the final output, the authors implemented six different models, which are variations of Markov Models and Neural Networks.

More recently, Horn et al. [1] developed a system – Visual Information Vases (VIV) – aiming at capturing and modeling creative inspiration. They define inspiration as the interpretation of concepts from one domain, and their translation into a different domain. VIV receives an image as input, and uses it as inspiration to create 3D-printable vases. The colors of the image are analyzed and used to generate a color palette, with the most predominant colors in the image, to a maximum of eight. The palette is used to calculate four characteristics: hardness, activity, warmth and weight. Together, these define the aesthetic profile of the image. A Genetic Algorithm is used to create the vase. The fitness function attempts to approach the aesthetic profile of the vase (the same four characteristics defined previously) to the one of the image.

2.2. Music Generation

Different types of algorithms can be used to generate musical artifacts. Papadopoulos and Wiggins [13] identified several of the most commonly used algorithms:

- Mathematical Models, used for example in the M.U. Sicus-Aparatus system [12];
- Knowledge-Based Systems, used by Oliveira and Cardoso in their Emotion-Driven Music Engine (EDME) system [6];
- Grammars, used by Steedman [9] in the generation of chord progressions;
- Learning Algorithms, used by Johnson and Ventura [2], in the previously mentioned system;
- Genetic Algorithms, which were used by Scirea et al. in the Scientific Music Generator (SMUG) system, in combination with Markov Chains [8].

The first four tend to generate artifacts that follow a certain genre, as they either are trained with songs in a specific style (which is the case with Mathematical Models and Learning Algorithms), or they implement and follow rules in the composition process (Knowledge-Based Systems and Grammars). On the other hand, Genetic Algorithms are able to create new and unexpected solutions, although they tend to lack structure, which contrasts with human behavior when composing music. Recently, some researchers have used Multi-Agent Systems to generate musical artifacts, Navarro et al. [5]. For inspirational systems, we have seen the use of Genetic Algorithms, Neural Networks and Markov Models in the literature [1, 2].

3. Identified Features

To create a bridge between the visual and music domains, we had to identify the elements that are readily available or are required from artifacts from each of the domains. The starting point for our system are digitalized images and the output is a midi file. From these elements we try to extract the emotional mood present in the image and map it to the elements that can contribute to the same emotional mood in the music domain. In the following subsections, we summarize what we have learned about both domains.

3.1. The Visual Domain

Several different elements can be identified in an image: color, lines, space, texture, shapes, among others. A human can easily look at an image, identify these elements, create his interpretation and try to come up with an understanding of the image as

a whole. For a computer, this is still a very challenging task. While it can easily identify the colors and lines, the task of actually recognizing objects, people and animals, understanding their relations among each other and how they are positioned is still a challenge in computer vision [10]. Although extracting the semantic meaning in an image would have contributed greatly to our work, this is not possible, so we have focused on creating a more abstract interpretation of the image by extracting features from the colors present in it. It should be noted that the interpretation of an image and its colors can be a personal matter and may vary according to the person's culture or background, among others. However, we can still try to create a global generalization of color symbolism [4], which most of us will recognize.

Color has three properties: hue, value and intensity. The first is the base color, and all the different hues can be found on the color wheel. The second indicates the lightness or darkness of a color. The last refers to the strength and vividness of a color. When combining these properties, thousands of different colors can be created. In figure 1, the twelve different hues can be seen. Figures 2 and 3 show how the value and intensity can change a color with a specific hue.

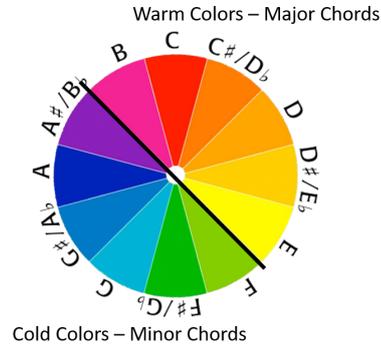


Figure 1: Crossover of the color wheel and the pitch wheel

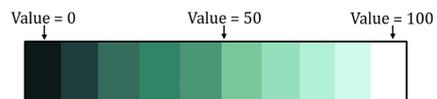


Figure 2: Value scale for the same hue

With this in mind, after obtaining all the pixels present in an image, we can extract different kinds of information:

- Individual pixels give us the specific color of each point in the image.



Figure 3: Intensity scale for the same hue

- A histogram of the image, that shows us most predominant colors of the image.
- How the color varies throughout the image, which allows us to understand if the image mostly maintains the same color, rarely changing, or if it constantly changes colors, creating contrasts.
- A categorization of all the colors into warm and cold, allowing us to understand if the image mostly transmits positive and happy emotions (warm colors), or if it tends to be more negative and sad (cold colors). Figure 1 shows this categorization.
- Groups of pixels which share the same color and are positioned next to each other, creating regions throughout the image.

3.2. The Musical Domain

Music is created by bringing together different types of sounds. The final result usually expresses some kind of emotion, and it may or may not be generally considered beautiful¹. Like described for the visual domain in the previous section, there are also many different elements in music, that can be used to create an emotional mood. The most basic musical element is a **note**, which has four characteristics: duration, pitch, intensity and timbre. By combining several notes together, we can create the main **melody** of a music.

The speed at which a music is supposed to be played is called the tempo, and is usually measured in beats per minute (BPM). Furthermore, a song is always divided into several parts, which are called bars. Bars always have the same duration, and the **time signature** determines how much that is. The rhythm of a music is defined by the tempo and the time signature.

A succession of notes is called a **scale**, and these may be ascending or descending. The name of a scale is always given by its first note, and it may be **(M)ajor** or **(m)inor**. When playing three, or more, notes simultaneously, a **chord** is formed. These can also be major or minor, depending on the notes that are present. By playing several notes together, **harmony** is created. This may be dissonant, if the sound is harsh, or consonant, if it is

¹The visual domain can also be evaluated aesthetically. However, we only use an image as a starting point. As such, its aesthetic value does not matter to us.

smooth. When playing in a major scale, or using major chords, the resulting song tends to have a happier mood, whilst minor scales and chords tend to result in more negative moods. Oliveira and Cardoso [6] identified some possible relations between several musical elements, and the resulting emotional mood. We have summarized some of these, and added some details, on Table 1.

	Happy	Content	Angry	Sad
Scale	Major	Major	Minor	Minor
Volume	Med-High	Med-Low	High	Low
BPM	Med-High	Med-Low	High	Low
Note Density	Med-High	Med-Low	High	Low
Note Duration	Med-Low	Med-High	Low	High

Table 1: Relation between the musical features and the emotional mood.

Finally, a song tends to have a **structure**, with specific instruments and elements in the song. This usually depends on the chosen genre, as some have very rigid structures, whilst others may not even have a defined structure. For example, classical music played by an orchestra uses many different instruments, whilst pop and rock music typically use three or four different instruments. Each genre specifies its own set of rules on how to combine the previously specified elements. It is possible to compose music by ignoring most of these rules, which results in the creation of music that may not be considered beautiful by the general public. On the other hand, by following the defined rules, the created music tends to be catchier and more easily liked. For our work, we have decided to generate music that somewhat follows the rules of the rock genre. This simplifies some important decisions, further discussed in the next section. Rock music usually has a drum beat, an electric guitar, an electric bass and sometimes a piano.

4. Cross-Domain Analogy

Having identified the relevant elements from each domain, they need to be mapped one to the other meaningfully. Since we have chosen to generate Rock music, we have:

- Rock music typically has between 90 to 110 BPM. However, it is possible to go both slower and faster than that interval. As such, we have decided to generate music between 50 and 150 BPM, which allows us to differentiate between extremely varied images and those that have very few color variations.
- A music usually has a main melody and, as such, we need to create a sequence of notes.

- Usually, there are chords played throughout a Rock music. We typically find progressions of three to four different chords being played in a music, in an electric guitar or on a piano.
- Finally, a drum track is imperative in the creation of rock music.

The first value to determine is the number of BPM in the generated music. We considered that an image with more color variations corresponds to a faster music, and vice-versa. As seen in figures 4 and 5, with low color variations there is a feeling of slow movement, whilst higher variations give us a fast-paced movement. As referred previously, a music is divided into several sections with the same duration, which is determined by the time signature. Rock music tends to be composed with a $4/4$ time signature, so we have determined that every artifact generated by our system belongs to that time signature.

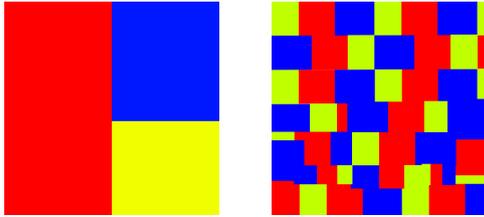


Figure 4: An example of low color variation

Another important mapping is that a color can give us a note. Considering that a color is defined by: (1) its hue – the color itself; (2) its value – its darkness or lightness; and (3), its intensity – how dull or vivid it is –, a direct translation can be made to the pitch (the note itself and its octave) and volume of a note:

1. First, the hue of the color is analyzed. There are twelve different hues in the color wheel, as there are twelve different notes in the chromatic scale. Each one of these hues can be directly translated into a note – for example, a red hue corresponds to a C. The chromatic scale can also be arranged into a circle, so the last note of a scale is also its first. Figure 1 shows how these two circles are overlapped.
2. Then, the register of the note should be selected. A higher register corresponds to a higher pitch, and vice-versa. The value of the color gives us the register of the note. So, continuing our previous example, if our red hue has a value of fifty, then our C note is in the fourth octave (generally written as C4).

3. Finally, the intensity of the color gives us the volume of the note. If a color is dull and grayed, then the corresponding note should have a low volume. Likewise, a bright and intense color corresponds to a higher volume.

However, a normal resolution image has far too many pixels to allow us to create a direct pixel-to-note translation, as this would generate very long artifacts. To mitigate this, we have condensed pixels that represent the same color and that are together in the image into "same colored regions". These regions are translated into one note, and its duration is directly related to the number of the pixels in the region. This process is explained in more detail in the next section. The analogy of translating a color into a note is also used to choose the chords in the music. For example, if red is one of the most predominant colors, then we have several C chords played throughout the music.

In the visual domain, we have discussed that colors can be categorized into warm and cold colors. Colors that belong to the former tend to feel more vivid and positive, whilst colors from the latter tend to give a more calm and negative feeling. Likewise, a major scale in music transmits positive and happy emotions, while a minor scale is usually more sad and negative. This results in an obvious analogy – if more warm colors are present, then the resulting music mostly uses major scales and chords; if more cold colors are present, more minor scales and chords are chosen. Figure 1 shows which colors are cold, and which are warm, and which chords are major and minor.

Finally, the information presented on table 1 can also be applied to the percussion of an artifact. As such, we summarized how the visual elements influence the emotional mood of an image on table 2. To create a drum beat for our artifact, we need to evaluate these elements of the image to understand its emotional mood. Then, the musical elements need to be selected to create the closest emotional mood possible. For example, if an image shows a happy mood – high warmth, a relatively high number of color variations, a high color intensity and value –, then the resulting drum beat should have a medium-high volume, BPM and note density, and a medium-low note duration.

	Happy	Content	Angry	Sad
Warmth	High	Med-High	Low	Low
Avg. Region Size	Med-Low	Med-High	Low	High
Color Intensity	High	High	Medium	Low
Color Value	Med-High	Med-High	Low	Med-Low

Table 2: Relation between visual features and emotional mood.

5. Implementation

The system was developed entirely in Python 3.5. To extract all pixels from an image, and to convert these to different color models we used scikit-image², a Python module for image processing. To generate the MIDI files, we used the Python module MIDIUtil³.

First, it is important to consider one major difference between the visual and the musical domains: whilst the latter has a specific order, a start and a finish, the former does not – an image tends to be seen as whole. However, a computer reads and processes the pixels in a given order. Whichever that order is it, influences the final results. As such, we give the user the possibility to choose the starting point and the order in which they are read. Currently, there are eight different choices, which include starting from the four different corners, and reading left to right, right to left, top to bottom or bottom to top. Figure 6 shows two examples of different starting points and reading orders.

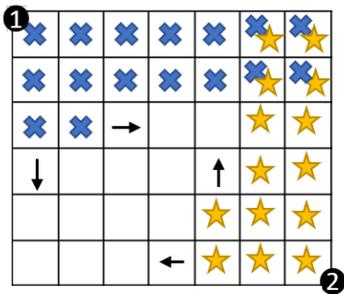


Figure 6: Image traversal examples.

A second concern to be taken into account is how much we should change the generated artifact to comply with a possible aesthetics function. On one hand, we can directly apply all the previously discussed analogies, making a direct translation of the image into a music. However, the resulting artifact may not be considered as music by people, since, in general, music harmony rules are not satisfied by the direct pixel translation. On the other hand, we can obtain an abstract profile of the image, such as the average warmth, average intensity, among others, use these to calculate the musical features, and restrict them to better follow harmonization rules. For example, if the most predominant color is red, then we should mostly use C major chords throughout the music. However, instead of using the second and third most predominant colors to choose the other two chords, we could choose F and G major, as C–F–G is a common rock progression, and it is a known fact that these three chords result in a

pleasant harmony. A question that arises, however, is if we can still consider this generated music as the music of the image, that is, if after applying some restrictions, the resulting music is still inspired by the features present in the image and can still be related to it. As such, we decided to create two implementations:

- A "raw" version, where a pure translation of the visual features into the musical features is attempted.
- A harmonized version, where the visual features are processed, taking into account the whole image, and where harmonization rules are applied in an attempt to "beautify" the resulting music according to the aesthetic characteristics of the chosen structure.

In this section, we explain how we implemented each one of the two analogies. When relevant, the differences between the raw and the harmonized versions are explained.

The first feature to be calculated is the number of BPM of the music. By using the CIE 2000 color-difference formula [3], it is possible to determine how many times there is a significant color variation in the image. To determine these variations, and then calculate the number of BPM we start by analyzing the image line by line, and each time a significant color difference is found we count a new region. When we reach the end of each line, the average size of same colored regions is calculated, by dividing the number of pixels in a line by the number of regions in that line. This is repeated for every line in the image. By considering all line we calculate the average size of same colored regions for every line in the image. A percentage of the average region size is calculated in relation to the total line size. This is repeated for the columns in the image. When we have the percentages for both lines and columns, an average is calculated between these two values. Finally, this value is translated into the number of BPM. The higher the average size of the same colored regions, then the less variations in the image. This results in a lower BPM value. This translation uses a linear function that maps the percentage value to the number of BPM. The domain of this function is $[0, 100]$, and the range is $[50, 150]$, since we generate music between these BPM values.

After having the number of BPM, the image is divided into several sections, all with the same size, called quadrants. Just like every bar in a $\frac{4}{4}$ music has four beats, each quadrant in our image corresponds to four beats. The following steps are taken to calculate the dimensions of these quadrants:

1. All the divisors of the height and all the divisors of the width of the image are calculated.

²<http://scikit-image.org/>

³<https://pypi.python.org/pypi/MIDIUtil/1.1.1>

This allows us to know into how many parts we can divide the image horizontally and vertically. The result is two sets of numbers. For example, an image sized 15 by 10 pixels has the following divisors, respectively: (1, 3, 5, 15) and (1, 2, 5, 10).

2. All the combinations between these two sets give us the possible dimensions for the quadrants. We are not interested in quadrants that are sized one, nor in quadrants that are the size of the image, therefore, in our previous example, this results in (3, 5) and (2, 5), where all possible dimension combinations are: (3, 2), (3, 5), (5, 2) and (5, 5).
3. To select the best combination, two criteria are followed: The resulting artifact should not be longer than three minutes, nor shorter than one minute, and the width and height of the quadrants should be as close as possible. The total number of quadrants is calculated by dividing the total number of pixels by the number of pixels per quadrant. The total duration of the artifact is calculated by multiplying the number of quadrants by four, as each quadrant corresponds to four beats.

Each of the resulting quadrants will have its own histogram from where we can extract several features:

- Most predominant colors in the quadrant.
- Average intensity of the colors in the quadrant.
- Average value of the colors in the quadrant.
- Average warmth of the colors in the quadrant.
- Same colored regions.

To generate the melody of the artifact, we use the basic analogy from color to note explained in the previous section. In our current implementation, there is no difference between the raw and the harmonized versions. Each same-colored region in a quadrant region generates a note, according to its color, and the duration of the note corresponds to the size of the region. So, for example, assuming we have figure 7 as a quadrant, then the following notes are generated:

- D \sharp , starting at beat 0, and with a duration of two beats;
- F \sharp , starting at beat 2, and with a duration of one beat;
- C, starting at beat 3, and with a duration of half a beat;

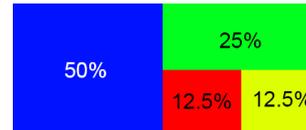


Figure 7: Example of a quadrant divided into same colored regions.

- G \sharp , starting at beat 3.5, and with a duration of half a beat.

After melody generation, chords are introduced in the music. One chord is played per quadrant, as each quadrant corresponds to four beats in the music. The most predominant color in each quadrant determines which chords are played throughout the music, similar to the analogy done in the melody generation. The warmth of that color determines if it is a major or a minor chord. Figure 1 shows the twelve different chords that can be chosen. There are some differences between the raw and harmonized versions:

- In the raw version, chords are selected individually by each quadrant. So, for example, if the most predominant color in the first quadrant is red, a C major chord is played in the first four beats. Then, if the second quadrant has blue as its predominant color, the next four beats have an E minor chord. This is done for every quadrant in the image.
- In the harmonized version, the maximum number of different chords is restricted to four, as a rock music typically uses no more than four different chords. Therefore, we count how many times each color is a predominant color in all the quadrants. For example, in an image with twenty quadrants, we can have a final count of ten for red, five for blue, three for light green and two for purple. The four colors with the highest count are selected, and then are semi-randomly distributed through the duration of the artifact. The chords corresponding to colors with higher counts have a higher probability to be selected, while the others have a lower probability. In our example, red corresponds to a C major and has 50% probability to be chosen, blue is a D minor and has a 25% probability to be chosen, light green is a G with 15% probability and yellow a G \sharp with 10% probability of being chosen. This could result in a myriad of different chord orders, which provides a non-deterministic aspect to our program. It should be noted that the final music, in this example, has 20 chords, since the image was divided into 20 quadrants.

Finally, to generate the drum track, we have created a knowledge base of 42 different drum beats used in several popular rock songs. A drum beat is a rhythmic pattern which is repeated throughout a music. Each one of these drum beats is four beats long. For example, figure 8 is a drum beat in our knowledge base, taken from the song "Hotel California", by the Eagles. The emotional value of each of these beats is calculated based on the features presented on table 1, and its selection is different for the raw and harmonized versions:

- If we are generating the raw version, then an emotional value is calculated for each quadrant, based on the features presented on table 2. As such, each quadrant generates a specific drum beat, which has the closest emotional value to that of the quadrant.
- In the harmonized version, we use the general emotional value of the image. It is calculated the same way as in the raw version, however the values are averages for the whole image. First, the closest drum beat is first selected, and is played throughout most of the artifact. However, every sixteen beats (which correspond to four bars in a music), the second closest beat is played, introducing some variety to the drum track.



Figure 8: Drum beat from the music "Hotel California", which is played at 72 BPM at a medium-high volume.

Formula 1 shows us how to calculate the emotional mood of a drum beat, where V is the beat volume, $NDen$ is the note density and $NDur$ is the note duration. Formula 2 shows us how to calculate a visual mood, where ARS is the average region size, W is the warmth, I is the color intensity and V is the color value. Both values vary between zero and 100, which makes them directly comparable. Both raw and harmonized versions use the same formulas, however the values used in formula 2 differ as explained above.

$$beat = 0.25 * (BPM + V + NDen + (100 - NDur)) \quad (1)$$

$$visual = 0.25 * ((100 - ARS) + W + I + V) \quad (2)$$

After each of these layers is generated, a MIDI file is created, and each layer is inserted into a different track. Both the chords and the melody are played in piano.

6. Genetic Algorithm

Genetic Algorithms provide an interesting solution to the generation of musical artifacts, even though they tend to lack structure and work randomly. However, it was our intention to maintain the possibility of creating novel and unexpected results, so we decided to generate a third version that resulted from the execution of a Genetic Algorithm. This type of algorithm divides itself into several phases:

1. **Initialization:** During this phase, which only occurs once in the execution of the algorithm, the initial population is generated. The population is a set of 24 individuals, where each individual corresponds to a musical artifact with a sequence of bars, chords and drum beats. This number was chosen due to the fact that it is generally considered that a population size between 20 and 30 is good. The first two individuals are the raw and harmonized versions that were generated previously. The remaining 22 individuals are generated from randomizing the chord sequence and the notes sequence in each bar.
2. **Elitism Factor:** To guarantee some level of quality in each new generation, some of the individuals are directly copied from the old generation into the new generation. Exactly one fourth of the current generation is copied – which corresponds to the best six individuals. The other 18 individuals of the new generation are created by selection, crossover and mutation.
3. **Selection:** Each pair of individuals is chosen by Roulette Wheel Selection – their probability selection is directly proportional to their fitness value. This means that higher fit individuals are more probable to be selected, but every single individual can be chosen, even if with less probability.
4. **Crossover:** On each pair of selected individuals there is a 90% chance to occur a crossover. Due to all the different elements in each individual, four different types may occur: (1) Chord Crossover, exemplified in figure 9; (2) Bar-Note Crossover, exemplified in figure 10; (3) Bar Crossover, exemplified in figure 11; and (4) Note crossover, exemplified in figure 12.
5. **Mutation:** Once again, several types of mutation can occur to both individuals: (1) Chord mutation, where a chord is randomly changed to any major or minor chord – 10% chance; (2) Beat mutation, where a beat is randomly changed to any other beat in the knowledge

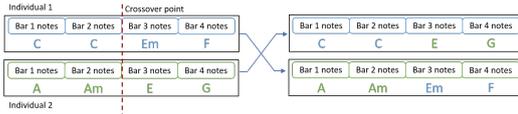


Figure 9: Example of a chord crossover.

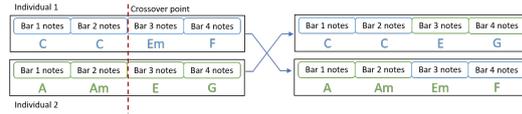


Figure 10: Example of a Bar-Note crossover.

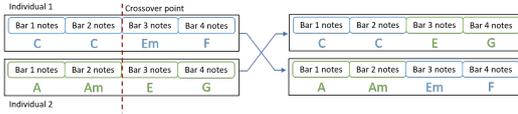


Figure 11: Example of a Bar crossover.

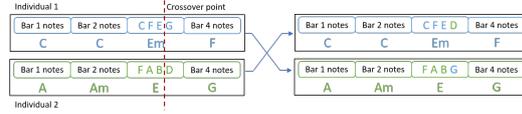


Figure 12: Example of a Note crossover.

base – 5% chance; (3) Note mutation, where one note per bar has a 15% chance to change either its pitch-class (50% chance), its octave (25% chance) or its duration (25% chance).

6. **Calculate Fitness:** The two new individuals have their fitness calculated, which is a sum of the seven measures presented on table 3.
7. **Finishing Conditions:** Finally, the algorithm will come to an end when 300 generations have been calculated. The final solution will be the most fit individual of the 300th generation. Similar to the raw and harmonized versions, a MIDI file will be created with the information given by this individual.

7. Evaluation and Discussion

A total of 15 artifacts were generated, blablabla, website

This project was divided into two main phases: the first, where the raw and harmonized versions were implemented, and a second, where the Genetic Algorithm was added. The artifacts generated in both phases were evaluated via on-line surveys. Between both phases, the bass layer was added. The first evaluation required people to evaluate *Broadway Boogie-Woogie* and *Girl Before a Mirror*, while the second evaluation required them to evaluate *Broadway Boogie-Woogie*, *Green and Maroon* and *Starry Night*.

Both evaluations - *i* is music is not rock :’(is liked. Three versions were preferred. Cenas

8. Conclusions Acknowledgements

The author would like to thank ...

References

- [1] B. Horn, G. Smith, R. Masri, and J. Stone. Visual Information Vases: Towards a Framework for Transmedia Creative Inspiration. In *Proc. of the 6th Intern. Conf. on Computational Creativity (ICCC 2015)*, pages 182–188, 2015.

- [2] D. Johnson and D. Ventura. Musical Motif Discovery in Non-Musical Media. In *Proc. of the 5th Intern. Conf. on Computational Creativity (ICCC 2014)*, pages 91–99, 2014.
- [3] M. R. Luo, G. Cui, and B. Rigg. The development of the cie 2000 colour-difference formula: Ciede2000. *Color Research & Application*, 26(5):340–350, 2001.
- [4] J. Morton. *A Guide To Color Symbolism*. Colormcom, 1997.
- [5] M. Navarro, J. M. Corchado, and Y. Demazeau. A Musical Composition Application Based on a Multiagent System to Assist Novel Composers. In *Proc. of the 5th Intern. Conf. on Computational Creativity (ICCC 2014)*, pages 108–111, 2014.
- [6] A. P. Oliveira and A. Cardoso. A Musical System for Emotional Expression. *Knowledge-Based Systems*, 23(8):901–913, 2010.
- [7] M. Rhodes. An Analysis of Creativity. *The Phi Delta Kappan*, 42(7):305, April 1961.
- [8] M. Scirea, G. A. B. Barros, N. Shaker, and J. Togelius. SMUG: Scientific Music Generator. In *Proc. of the 6th Intern. Conf. on Computational Creativity (ICCC 2015)*, pages 204–211, 2015.
- [9] M. Steedman. A generative grammar for jazz chord sequences. *Music Perception*, 2:52–77, 1984.
- [10] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2010.
- [11] T. M. Thrash and A. J. Elliot. Inspiration as a Psychological Construct. *Journal of Personality and Social Psychology*, 4(84):871–889, May 2003.
- [12] J. Toivanen, H. Toivonen, and A. Valitutti. Automatic Composition of Lyrical Songs. In

Fitness Measure	Case	Value	
Chord Progression	Each time three sequential chords form a Rock progression.	(+25)	
	Each time two sequential chords form a Rock progression.	The third chord is typically found in Rock.	(+15)
		The third chord is not typically found in Rock.	(+10)
	Each time three sequential chords do not form a Rock progression.	The three chords are typically found in Rock.	(+5)
		Only two chords are typically found in Rock.	(+0)
		Only one chord is typically found in Rock.	(-10)
	None of the chords is typically found in Rock.	(-25)	
Chord Repetition	Each time three sequential chords are the same.	(-25)	
Note Repetition	Each time two sequential notes are the same.	(-15)	
Long Notes	Each time a note has a duration of two or more beats.	It is a chord note.	(+15)
		It is a scale note.	(+5)
		It is a rest.	(-15)
		None of the previous cases apply.	(-10)
Downbeats	For every note in the first beat of a bar.	It is a chord note.	(+15)
		It is a rest.	(+10)
		It is a scale note.	(+5)
		None of the previous cases apply.	(-10)
Range	If the range of the song is lower than 18.	(+10)	
Intervals	For each interval that is larger than an eighth.	(-20)	

Table 3: Fitness value measures

Proc. of the 4th Intern. Conf. on Computational Creativity (ICCC 2013), pages 87–91, 2013.

- [13] G. A. Wiggins, G. Papadopoulos, S. Phon-Amnuaisuk, and A. Tuson. Evolutionary Methods for Musical Composition. In *International Journal of Computing Anticipatory Systems*, 1999.