The Image Artist: Computer Generated Art Based on Musical Input

Paper type: System Description Paper

Viktor Zorić and Björn Gambäck
Department of Computer Science
Norwegian University of Science and Technology
7491 Trondheim, Norway
viktor.zoric@outlook.com gamback@ntnu.no

Abstract

The paper explores computer-generated art based on musical input, using evolutionary algorithms (EA) for a music-to-image transformation which operates on music and image metadata, such as music and image features, rather than raw data. The metadata is utilized through a novel usage of mapping tables that work as recipes for images. The mapping tables are evolved without interactive involvement. Experiments were carried out on using the mapping tables to match music and image features, and with various fitness functions that combined user preferences with art novelty aspects. Fitness functions based on a distance measure between user preferred mapping tables and the computer-generated mapping tables can efficiently drive the EA towards the user's preference. Novel and interesting mappings between the image and music features can be achieved with fitness functions that selectively ignore some user preferences. Evolution of palettes and figure parameters can match user expectation without knowing the structure of the optimal mapping table.

1 Introduction

Visual art can enrich various aspects, phases and situations in human life, with the support role for computing art being situation dependent. The topic of this work is to create a system that can generate image art based on musical input with use of evolutionary algorithms (EA). This means that the system should create images that share features with the corresponding music. Such features could be emotional or artistic, where the aim is to create a correlation between image and music that the end user agrees on. The system should be able to take an arbitrary musical piece as input and create a corresponding image, considering both end user specifications and novelty.

The work combines a theoretical and a practical approach. It is a design, implementation and experiments-driven exercise, where end-user involvement — survey results and user interaction tests — contributed to the EA functionality. The EA for music-to-image transformation operates on music and image metadata (attributes), rather than raw data. The metadata (for music and image features) is utilized by mapping tables that work as recipes for images. The algorithm generates images by evolving the mapping tables without interactive involvement. Using metadata and mapping tables in evolutionary algorithms introduces an alternative approach to computer generated image art, compared to previous research.

The next section introduces some work that has inspired the present project. Then Section 3 describes the system architecture, while Section 4 shows some experiments using the system. In Section 5 a discussion of the system is presented, and possible future work outlined.

2 Related Work

Over the last two decades there have been many different approaches to generation of art using computers, with evolutionary algorithms being a recurring method. EAs are highly dependent on a fitness function which accurately describes, in mathematical terms, how good a solution is. Lacking this feature the algorithm will struggle to generate a good solution set. But evaluating aesthetics and art is a subjective process, so a well performing general mathematical fitness function for art is absent. Instead of writing functions that find subjectively good looking patterns in image art, several approaches to generative art programs thus use interactive search methods, where humans take part in the evaluation of aesthetics/quality. In interactive evolutionary algorithms (Sims, 1991), humans must evaluate solutions through subjective judgement (Eiben & Smith, 2015), that the algorithm can use to generate offspring, by setting the fitness value of each solution, or by selecting phenotypes to mate. Following Todd & Latham (1992) interactive evolutionary computing dominated the evolutionary art field in the 1990's, with the vast majority of the 200 citations cataloged by Lewis (2008) using some form of case-by-case human judgment.

Ashmore & Miller (2004) stress that the main difference between evolutionary art and other search problems is that the fitness of an image is based on something that is very hard to describe or maybe even to understand, since the attractiveness of an image is personal and differs among people. With evolutionary art, the search is more exploratory, with divergence and diversity being the key factors. Understanding the nature of visual representation requires asking what artists need to know in order to make representational objects; knowledge not only about the world, but also about the nature and the strategies of representation (Cohen, 1988). Woolley & Stanley (2011) showed that the used representation has a major impact on the evolution of images (including performance). Given the hardness of this kind of application, it would be desirable to have representations that have high locality (Galván-López et al., 2011), so that small changes to the genotype correspond to small changes to phenotype. Johnson (2016) classified a large collection of research using a taxonomy of the ways in which fitness is used in EA art and music systems, with two dimensions: what the fitness function is applied to and the basis by which the function is constructed.

Significant here are the analyses of Machado & Cardoso (2002), Baluja et al. (1994) and Kowaliw et al. (2009) that present various techniques to overcome the limitations of interactive EAs. Secretan et al. (2011) and Clune & Lipson (2011) use web-based interactive EA to let users evolve lineages of artifacts based on their preferences, rate evolved artifacts, and take previously evolved artifacts and continue the search process themselves, so that artifacts can be the product of a collaborative search process. The present work will try to make a compromise by using the results of both user interaction and automated computing based on fine-tuned fitness functions to steer evolutionary algorithms. This complementarity can possibly offer both promising artistic results and convergent algorithms.

One of the key aspects of the evolutionary art is the novelty. Lehman & Stanley (2010) proposed a novelty search algorithm for evaluating image uniqueness. For each image, a novelty score is computed, taking into account its neighbours and an archive containing the most novel images. Vinhas et al. (2016) explore the effects of introducing novelty search in evolutionary art (they define novelty as phenotypic diversity). Their algorithm combines fitness and novelty metrics to frame image evolution as a multi-objective optimization problem, promoting the creation of images that are both suitable and diverse. Dumoulin et al. (2016) investigate the construction of a single, scalable deep network that can capture the artistic style of a diversity of paintings. They claim that their model permits a user to explore new painting styles by arbitrarily combining the styles learned from individual paintings.

Two projects have particularly inspired the present work: *The Painting Fool* (Colton, 2012) creates art by simulating natural painting strokes of different types through parameterization, which allows for the discovery of novel painting styles, combined with a database of mappings between emotions and different styles, to alter some styles by enhancing given emotions. Evolutionary algorithms are also implemented to expand the abilities to create novel results. *Sågawave* (Bredesen et al., 2016) focuses on creating images from songs, using Spotify API to fetch songs, Web Audio API to analyze them, and React front end library to draw images. Images are generated while the music is playing, and drawn from left to right as the songs progress. Frequency values determine how many shapes there will be and where on the canvas they are drawn. Amplitude values are used to select shape colours, while number of beats per minute map to weighting of colours and whether to draw sharp edged objects or not.

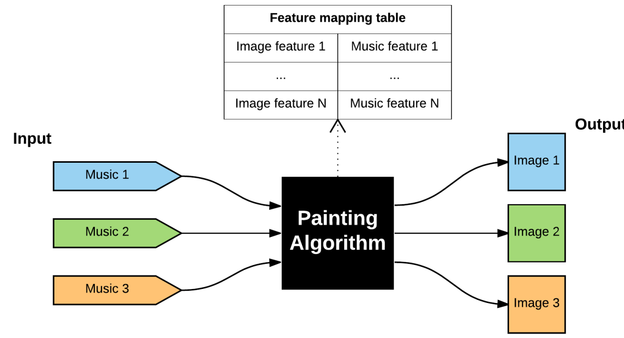


Figure 1: The Workflow of the Painting Algorithm

3 Architecture

This section discusses the architecture and design of the system for Music-to-Image transformation using Evolutionary Algorithms. The EAs are implemented from scratch to have full control of the evolution and its strategies. The whole framework is written in Java, utilizing Java's built in graphical library, and with a custom written interface for fetching music features using Spotify API. Spotify is a music streaming service that provides access to millions of songs that have downloadable previews containing audio analysis data.

Evolutionary Algorithms depend on three main parts: the genotype, the phenotype, and the evolutionary loop. The evolutionary loop is further guided by a fitness function. Creating images from music requires a mapping between music and image features. Music features are obtained through Spotify API, while image features need to be created during the act of painting using available painting tools. Figure 1 illustrates the workflow of the painting algorithm. The **phenotype** in this work will be called "The Image Artist" and produces a set of images, based on the **genotype** (represented by the feature mapping table). For each input song, a corresponding image is generated by using this feature mapping table. It is a metadata set (with key-value pairs) used to create painting instructions.

The evolutionary algorithms use the music indirectly, i.e., via various music parameters / descriptors, also denoted as metadata. Rather than running audio analysis on raw data, which requires the client to possess the music file, Spotify API can be used to obtain audio metadata that contain more information than the local functions currently can return. Spotify audio features objects, obtained using the API, contain several variables to describe the song: duration in milliseconds, key, mode (major or minor) loudness (in decibel), tempo (beats per minute), time_signature (number of beats in each bar), energy (a perceptual measure of intensity and activity), "danceability" (how suitable a track is for dancing based on a combination of elements such as tempo, rhythm stability and beat strength), "instrumentalness" (prediction of whether a track contains no vocals), "acousticness" (a confidence measure of whether the track is acoustic or not), "liveness" (detection of the presence of an audience), "speechiness" (detection of spoken words), and valence (how positive/glad or negative/sad the track sounds).

Various image parameters, obtained either in the preprocessing phase (i.e., while generating the image) or at the postprocessing phase (image analysis of the finished image) can be used to characterize the images. Image metadata is implemented as an enumeration of tool parameters. This enumeration tells the panting algorithm how to parameterize each painting tool. This set of parametrized tools gives a direct description of how the resulting image will look. However, for another type of image analysis, such as pattern recognition or search for other hidden image features, postprocessing is required. One such postprocessing function has been implemented, which extracts a colour palette from the image using the k-means clustering algorithm. The returned colour palette can be closer to the perceived colours in the image than the original palette used due to colour mixing during painting. Some evolutionary art projects use image analysis for evolution (Machado & Cardoso, 1998; Klinger & Salingaros, 2000). The present framework contains functionality to evolve raw images. This means that the phenotypes in the EA are images, and that the fitness functions directly analyze the images. The system uses evolution on metadata, and analyzes the parameters used to create the end result rather than analyzing the end result itself.

Formally, a **mapping table** $t \in T$ (where T is the set of mapping tables that the painting algorithm utilizes) is a function used for feature mapping, utilizing image parameters as keys (K) and music parameters as values (V): $t = f : K \to V$. An image r is created by adding functions of music parameters. Such a function of the music parameters can be denoted as a painting tool. "The Image Artist" uses several tools ($f_1 \dots f_k$) to create an image r, each of them being a function of the music parameters

$$r = f_1\left(p_1^m, ..., p_i^m, ..., p_N^m\right) + \dots + f_k\left(p_1^m, ..., p_i^m, ..., p_N^m\right)$$

where $m = \left(p_1^m, ..., p_i^m, ..., p_N^m\right) \in M$ define the music parameters of a music file m belonging the set M of music files on which the painting algorithm operates.

The purpose of the **genotype** is to create a recipe for the painting algorithm that describes how music feature values are mapped into painting tool parameter values, e.g., musical tempo can be mapped to amount of brush strokes to paint, so that slow melodies create calm images, while high tempo melodies create chaotic images using lots of strokes. Hence the genotype can contain a mapping between the image feature 'brush strokes' and the music feature 'tempo' with a scaling interval $[20, 300]$, or a mapping between 'base colour' and music 'energy' with an interval $[270, 0]$.

The **phenotype** is an artist object ("The Image Artist"), which utilizes a mapping table representing the genotype. The task of the phenotype is to create an image recipe that can be used to paint the final image. For each tool parameter (key in the mapping table), the associated value is fetched. The value of the music parameter is used to calculate an output tool parameter value by linearly scaling the music value to the output interval.

Phenotypes are evaluated by a **fitness function** which uses subfunctions that estimate various image criteria. The number of subfunctions depends on the evaluation criteria and given goals of the image creation process. Examples of criteria can be user-defined aesthetical fitness, novelty function, and their combination. Optimum fitness is reached when the distance between the current genotype and an optimum mapping table is zero. As detailed below, three fitness functions have been implemented: optimizing towards a user specified mapping table, novelty combined with a user specified mapping table, and optimizing towards user preference without knowing the mapping table.

The "Optimizing towards a user specified mapping table" fitness function guides the evolution to find a mapping table that is "close" to what the user has specified. The distance between any two mapping tables is the sum of distances between key-value pairs in the mapping tables. Each tool parameter (key) should map to the correct music feature variable (value) and have the correct output interval. The Image Artist uses the output interval to calculate a value for a painting tool parameter. Given a target interval $T = [t_1, t_2]$ and current interval $C = [c_1, c_2]$, the distance between the intervals is $d\,(T, C) = |t_1 - c_1| + |t_2 - c_2|$. For mismatching music variables ($m_1 \neq m_2$) for a given tool parameter, the distance between the intervals is multiplied by a penalty factor k. The fitness function for the current genotype G is then calculated as a sum of contributions from all tools:

$$f\,(G) = \sum_{i=TP_1}^{N_{TP}} (d(T,C)k(m_1, m_2))_i$$

where N_{TP} is the number of tool parameters.

The fitness function "Novelty combined with user specified mapping table" creates a notion of novelty in the evolution, by optimizing towards a user suggested mapping table but ignoring some of the map entries and letting the system stochastically select how those tool parameters are mapped to music parameters and output intervals. Stochastical selection of parameter values is done through the nature of EA by not calculating fitness values for some parameters, therefore allowing any values for these parameters to propagate in the evolution. The EA uses the same distance and penalty functions as the previous fitness function, but does not iterate over all possible tool parameters: Some arbitrary tool parameters are not included in the fitness calculation, so some differences between the user selected mapping and the generated one will not be calculated in the fitness value. This fitness function can ignore *(a)* whole entries based on a key (tool parameter), *(b)* one mismatch between tool and music parameters, or *(c)* differences between output intervals. Hence user suggestions (representing user's aesthetic criteria) and novelty can be combined, and hopefully provide certain aesthetic qualities. The user can flag parts of the mapping table that the system can explore within. Ignoring user specifications leaves the system to arbitrarily select variables and values to use, giving it the possibility to introduce novelty in the results.

"Optimizing towards user preference without knowing the mapping table" operates somewhat differently compared to the previous two fitness functions. Instead of having an optimum mapping table to optimize towards, this fitness function is guided by user descriptions of how the final result should be. The fitness function analyzes how the mapping table affects the image in each genotype and compares these results with the provided information for each song. The user can, e.g., specify how the final colour palette should be without stating how the palette should be generated. Thus, the system will be missing information about critical parameters, and must find a mapping table that can generate the requested final result. This guides the evolution to search for mapping tables that match requested end results rather than predetermined mapping tables, so that the evolutionary algorithms can introduce interesting and unexpected mapping tables. The fitness function leverages a distance measure $d(A,B) = |A-B|$, where A is the user requested result and B the currently generated result. Depending on the opinion of the user, A and B can have different meanings, varying from colour palettes to the total number of brush strokes. This fitness function can introduce unexpected mapping tables that match user preferences but have interesting effects with other music. It can also be combined with one of the previous mentioned fitness functions, such that concrete user preferences can be combined with abstract preferences.

Furthermore, the system allows for varying the evolutionary algorithm's selection strategies, crossover techniques and mutation. Two **selection** strategies and two **crossover** alternatives have been implemented, proportionate selection and tournament selection resp. One-Point crossover and Uniform crossover (see, e.g., Floreano and Mattiussi, 2008). One-point crossover slices two mapping tables at an arbitrary index and combines the two parts from each genotype into a new genotype. Uniform crossover iterates over all keys and stochastically selects which value from the two tables to duplicate into the new genotype. **Mutation** is an essential part of the evolution, which is necessary to introduce diversity among the population and ensure a more complete search in the domain space. Mutation techniques can be modelled as stochastic processes that influence offspring. Having a mapping table as genotype, a new mutation technique must be implemented such that all parts of the table are mutable. This means that key-value pairs can be altered, and the information within the values can be modified. With a given probability, the feature variable is altered, such that selected tool parameter (key) is mapped to a different feature variable, or the output range is altered using a given mutation pressure. The mutation pressure in an interval $[-t, t]$ from which a random value in this interval is selected and added to a numeric variable selected for mutation.

The **painting algorithm** as such is mainly for creating abstract art by using different shapes and simulated brush strokes. However, it is not limited to abstract art: having simulated brush strokes allow for the creation of many art styles. A range of painting tools (for shapes, brush strokes and image effects) can be combined using layers, where each tool creates a layer on top of a digital canvas. The tools are highly parameterized to utilize each to its full potential.

Three main **geometric shapes** have been implemented: rectangles, ovals, and polygons. Rectangles can segment an image into sections or represent some objects. A combination of multiple rectangles in specific positions on the canvas can be used to create representations of more complex structures. Ovals are also useful for the representing objects, but since ovals have no edges, they can be used to enhance a calm emotion, a smooth motion or a soft object. Polygons are painted using random colours biased towards red. The number of points to be used, the positions of them, thickness and border colours are all parameters that can be set. Having many random points often yields pointy objects that can be related to aggressiveness and anger.

Two types of **brush strokes** are implemented: curved and straight. Curved strokes try to simulate brushes with a circular head, while straight strokes simulate brushes with a flat rectangular head. Simulating brush strokes can help the created images look creative, since humans can relate to results from human artists. Both types of brush strokes are implemented using a high number of regular straight lines, all following the same direction (from start of the stroke to the end). Every line within a stroke is altered differently as the stroke is painted to give the effect of paint smudging and fading. In the straight brush strokes, the colour intensity degrades as the stroke is painted, and fades out at the end. In the curved strokes, this effect is slightly reduced as it naturally occurs due to the layout of lines. The curved brush has all its painting lines in a 2D normal distribution, while the straight brush has lines evenly spread out among its width.

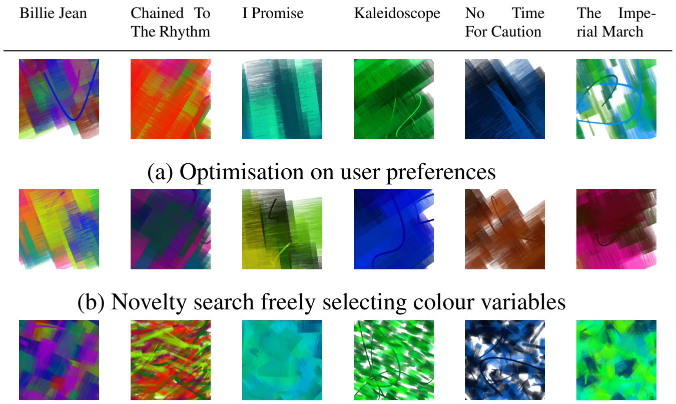
Currently there are three types of **image effects** implemented: cloud, blur and oil. The cloud effect mostly affect the colours of an object by creating a monochrome layer of noise that can induce some diversity among equal shapes. The blur effect softly smooths out sharp corners in an image, while the oil effect removes some of the clearly artificial lines resulting from the brush painting algorithm, so that the resulting image reminds of an artistic effect involving water or oil, rather than computer generated curved lines.

4 Experiments and Results

Various experiments were performed to validate system behaviour and to explore the importance and sensitivity of various techniques and approaches. Table 1 shows a basic experimental configuration. The population size is limited due to limited computational resources and time. A set of six songs was selected for these experiments. They differ from each other on several musical features, but are also equal in some features: *Billie Jean*, Michael Jackson; *Chained To The Rhythm*, Katy Perry, *I Promise*, Alex Kozobolis; *Kaleidoscope*, Kasbo; *No Time For Caution*, Hans Zimmer; and *The Imperial March (Darth Vader's Theme)*, John Williams.

Table 1: Basic EA configuration for the evolutionary runs

EA option	Value
Population size	20
Max generations	2000
Elites	1
Crossover rate	0.7
Mutation rate	0.7
Parent selection	Tournament
Crossover type	One Point
Mutation pressure	20



(a) Optimisation on user preferences

(b) Novelty search freely selecting colour variables

(c) Novelty search ignoring some user-defined keys

Figure 2: Evolution towards a user-defined mapping table

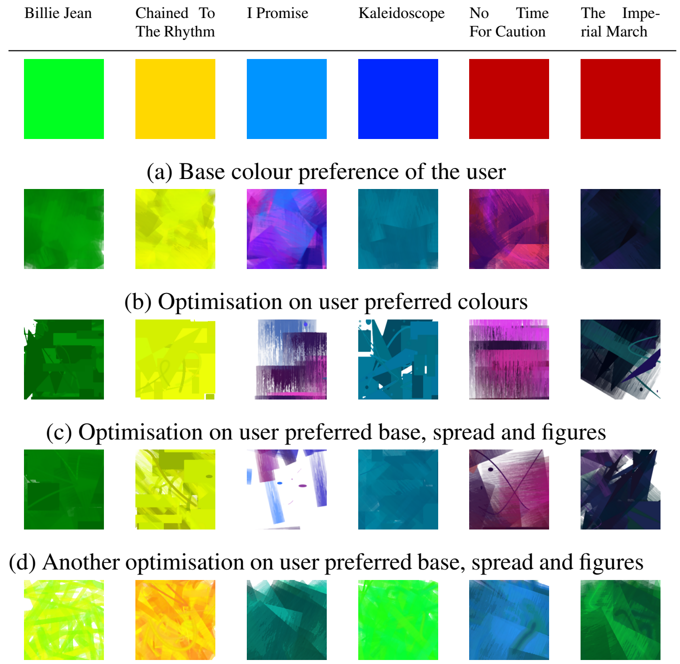
Experiments with fitness functions

Experiments with different fitness functions were performed to investigate how they influence end results and whether some parts of the function can guide the EA to fulfill some objectives. A small change in the fitness function can introduce novelty in the results or guide the EA directly towards user preference. The fitness evaluation can be based on the mapping table or the painted images.

The first experiments used the fitness function "Optimizing towards user specified mapping table", by first evolving a mapping close to the user's suggestion. Figure 2a shows the generated images after evolution. The colour palette is generated using the music feature variables energy and valence. "Chained To The Rhythm" has high energy and yielded a colour palette based on the colour red, while "I Promise" has low energy and got a colour palette in the blue spectrum, which fitted well with what the user had specified.

Figure 2b shows images generated by evolving towards a user preferred mapping table, but ignoring user specified parameters for colour, allowing the system to freely select the colour variables. Comparing Figure 2a and Figure 2b, the major difference is in the colour palette in each image. The "Billie Jean" image in Figure 2b has multiple bright colours, while in Figure 2a the colour palette is darker; however, the user claimed that both colour palettes fitted the music of "Billie Jean". For "The Imperial March", the user thought the image in Figure 2b fitted the music better than the one in Figure 2a, due to the presence of dark and red-pink colours. This was surprising and appreciated by the user.

A third experiment ignored some arbitrary parameters in the user preferred mapping table, to see whether the resulting images can surprise the user, while matching most of their preferences. Another purpose was to see how sensitive the system is to the parameters. The generated images introduce style variance by mostly differing from the previous ones in shape construction, with Figure 2c using a high number of small shapes, while Figure 2a uses few big shapes. It is visible that the system is sensitive to changes in parameter values. The user agreed that the images in Figure 2c fit to the music, but also introduce a positive element of surprise.



(a) Base colour preference of the user

(b) Optimisation on user preferred colours

(c) Optimisation on user preferred base, spread and figures

(d) Another optimisation on user preferred base, spread and figures

(e) Optimisation towards user preferred base, spread and figures, with seeding genotypes taken from earlier experiments (Figure 2a)

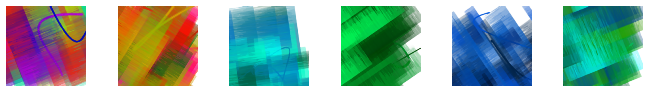
Figure 3: Evolution without a known mapping table

The set of experiments shown in Figure 3 optimize towards abstract user preferences without knowing the mapping table, meaning that the user specified how parts of the end result should be, rather than how to generate them. This evolution did therefore not have a known mapping table to optimize towards, but had to search for a mapping table fitting the user preferences. These preferences were extracted from a user survey, where for each song a base colour is selected, as well as image aggressiveness, and amount of brush strokes to use. The following experiments are based on the preferences from one arbitrary user, displayed in Figure 3a.

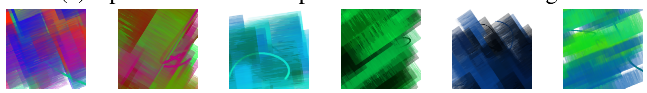
Figure 3b shows the results after evolution optimizing towards user preferred base colour for each song, and the colour spread in the palette. As there the amount of brush strokes is not optimised, the final amount happened to be high, so filling the whole painting canvas. Comparing the results to the user's preferences, there are some differences in shades, but there is agreement on the base colours.

Figure 3c shows the results after evolution optimizing towards the user preferred base colour for each song, and the colour spread in the palette. However, this set was generated using all the available painting tools, to generate a set of images that differ from other experiments. Comparing Figures 3b and 3c, there are two different painting styles in the images. The images in Figure 3c are more dynamic, with the use of several painting tools. The polygons provide some aggressiveness to the images, while the small ovals give elements of surprise that contribute to novelty. Figure 3d is another set generated the same way as Figure 3c, but with different parameters. This image set is slightly more dynamic with the use of rotation in some brush strokes.

Billie Jean	Chained To The Rhythm	I Promise	Kaleidoscope	No Time For Caution	The Imperial March

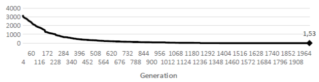


(a) Optimisation on user preference without seeding

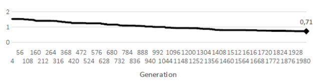


Billie Jean	Chained To The Rhythm	I Promise	Kaleidoscope	No Time For Caution	The Imperial March

(b) Optimisation on user preference with seeding



(c) Mean fitness without seeding



(d) Mean fitness with seeding

Figure 4: Experiments with seeding techniques

Figure 3e shows a result set after seeded evolution. The fitness function operates on preferences from the same user as in Figure 3b, but the evolution is seeded with genotypes from the experiment that produced Figure 2a. These two users have different preferences. Comparing Figures 2a, 3b and 3e shows similarities in all results, where Figure 3e shares image features from both experiments. This shows that seeding affects the images, and can introduce novelty.

The experiments in Figure 2) and Figure 3 used data from different users with different preferences, so the final results cannot be directly compared. However, all experiments did match user preference either through direct mapping tables or through specific requirements within the end results such as colours.

Experiments with seeding techniques

Seeding techniques are used to influence the initial population of the EA, to affect where in the search space the EA should start. Seeding has been met by scepticism (with, e.g., Eiben & Smith, 2015, claiming that it might be unnecessary), but can give the EA a push in the right direction Seeding can also be used to incorporate previously generated results matching user criteria, so that the EA can explore a local search space. These experiments aimed to investigate how seeding influences the results and EA performance, by (i) seeding initial population based on given genotypes, so that the search starts in a predetermined place in space, and (ii) initialize with fully stochastic population (no seeding).

All experiments were run five times, with the results averaged, and performed to observe how seeding affects both the evolution of mapping tables and resulting images. Evolution was optimized towards user specified mapping tables. Figure 4c shows the mean best fitness values without seeding. The steepest decline in fitness happens in the first 1000 generations. This experiment optimizes towards a user preferred mapping table, so it is expected to get results similar to Figure 2a. Figure 4a shows that this is indeed the case: the colours used are similar, as is the rotation of figures and amount of brush strokes. The visual differences are due to stochastic painting order and colour selection.

Figure 4d indicates that seeding drastically improves the performance of the EA in the first generations, as rediscovery of previous genotypes with low fitness values is avoided. However, after $2,000$ generations only slight improvements are noticable. Comparing the resulting images with (Figure 4b) and without (Figure 4a) seeding, it is visible that they share the same features. Figures 4c and 4d highlight the last best fitness value in each experiment, showing a mean improvement of only 0.82 across the two experiments (1.53 resp. 0.71). As Eiben & Smith (2015) stated, seeding is not necessary. The EA will eventually reach its target fitness value if configured correctly. However, if the objective is to reduce execution time, seeding can be an efficient option.

Experiments with user involvement

As the aesthetic judgement is subjective it is necessary to involve humans in the learning and evaluation process. Here, users were involved through a small survey and through interviews with one or two persons. The interviews were used to generate mapping tables the system can optimize towards. The questions were about relations between tool parameters and music variables, as well as numeric intervals. The survey was created to obtain a more general overview of user expectations on how the images should look, and to get feedback on the overall aesthetics of the generated images and how well they match the input music.

The user feedback presented below is a combination of individual responses and a summary of all users' responses. User expectations of how an image should look after listening to a specific song were described by four categories:

1. **Base colour** for palette generation, taking the user's response colour and making it darker or lighter if requested.
2. **Energy:** A measure from 1–5 where 1 is relaxing and 5 is aggressive. This scale is used to get an indication of how figures and brush strokes should be placed in the image.
3. The number of **brush strokes** to use in the image, on a scale from 1–10, where 1 is very few and 10 is many.
4. Expected colouring where the used **palette** should have:
   O1: One colour with small changes in shades
   F1: Few adjacent colours with small changes in shades
   M1: Many adjacent colours with small changes in shades
   O2: One colour with high variance in shades
   F2: Few adjacent colours with high variance in shades
   M2: Many adjacent colours with high variance in shades

To exemplify, Table 2 shows the user expectations for *Billie Jean*, *No Time For Caution* and *The Imperial March*, while Figure 5 summarizes the user feedback on the actual produced images for these songs (shown in Figure 2a). Most positive comments on *Billie Jean* related to the colours (the palette, the repetion and the relationship between colours), the amount of brush strokes, and that the image follows the rhythm. The negative comments included that it was too uniform, dark and geometrical, and had too many colors and exposed canvas. The image generated for *No Time For Caution* was mainly liked by the users, with comments that it was aesthetically pleasing, reflected the mood of the music, and had the right colours and colour temperature. The few negative responses said it was a bit too dark and needed more aggressive colours. On the other hand, *The Imperial*

Table 2: Individual user expectations

(a) Billie Jean

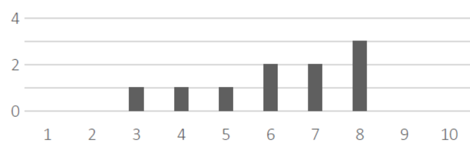
User	1	2	3	4	5	6	7	8	9	10
Base										
Energy	3	2	4	2	4	3	4	3	4	4
Strokes	7	3	8	5	9	5	8	7	7	8
Palette	M1	F1	M2	F1	F2	F1	F2	M1	F2	M1

(b) No Time For Caution

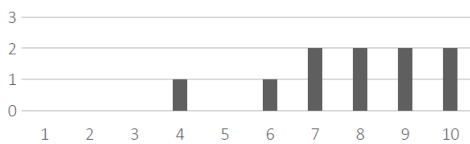
Base										
Energy	2	3	5	3	2	3	5	1	4	5
Strokes	1	6	9	7	2	5	9	2	8	6
Palette	O1	F1	F2	F2	F1	O1	F2	F1	F2	F2

(c) The Imperial March

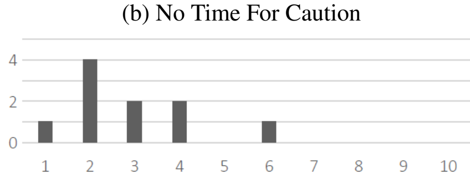
Base										
Energy	5	4	5	4	4	4	5	4	5	5
Strokes	10	8	10	4	6	6	9	8	10	8
Palette	F2	M1	F1	F1	F1	M1	M2	F1	M2	F2



(a) Billie Jean



(b) No Time For Caution



(c) The Imperial March

Figure 5: User feedback summary

March image was mainly disliked by users, who thought it had the wrong colours and mood, and was not aggressive enough. Still, the image got positive feedback on its small brush strokes, and for being dynamic and creative.

5 Discussion and Conclusion

This paper explores computer-generated art based on musical input, focusing on the use of evolutionary algorithms in image generation. The end-user involvement, survey results and user interaction tests, contributed to the system design of the evolutionary algorithm functionality, e.g., seeding techniques, feedback functions, and mappings between genotype and phenotype information. The evolutionary algorithms for music-to-image transformation operate on music and image meta-data, rather than raw data (music and image media). This requires a good metadata structure and organization, as well as good solutions for metadata and parametric representation of music and images. Some of the design achievements that should be emphasized are: *(a)* metadata-driven design of the genotype, *(b)* metadata-driven "coupling" between the EA and the painting tools, and *(c)* generic design of the fitness function (that gives a possibility to experiment with various fitness evaluation approaches).

As aesthetic judgement is subjective, it is difficult to create an automated evaluation process. Experiments with fitness functions and user involvement showed that the system was able to find a mapping table that is very close to the user preference, but that the resulting images sometimes were not optimal considering user expectations. The most important design tests and experiments focus on *(a)* using mapping tables to match music and image features and *(b)* various fitness functions that combine user preferences with art novelty aspects. The obtained experimental results and the end user interaction tests and evaluations are promising and motivate for further development of the tool set, as well as fine-tuning of the evolutionary algorithms.

The system can partially learn about user preferences through earlier experiences. The best genotypes after each evolution can be stored and reused through seeding. The key elements of one or more user preferences can then be collected through data analysis. The system can accurately reuse (single) user preference data through the seeding. The experiments show that the results may come from the evolution optimizing directly towards user preference, but also from fitness functions optimizing for novelty. The novel results can be approved by the user and included in the experience data. Diverse user preferences make it difficult to generate a mapping table fitting every user's preferences, so currently the system can at best learn individual user preferences. So far, not enough data analysis modules have been implemented to take full advantage of earlier experiences. Intelligently merging previously generated genotypes (based on how different music and image parameters affect end results and sensitivity) could produce more accurate solutions.

Assuming that the provided user preference through the metadata is accurate enough, the system can create images that are both æsthetic and meaningful. In some cases, there was some negative feedback on the image æsthetics and the music match. This is mainly due to subjective preferences of image quality. Even though users are not directly involved in the evolution, some user interaction can be introduced through the seeding. The seeding could be introduced mid-evolution to push the evolution in a specific direction (by choosing the genotypes with specific/wanted properties).

Due to subjective judgement of æsthetics and cultural differences, the system cannot create æsthetically pleasing and meaningful images without any user involvement, as also pointed out by Galanter (2010). This system thus involves end users in the initial stage of the evolutionary algorithms to obtain some guidelines towards user preferences. The system is able to generate pleasing images for end users that share at least some notions of aesthetics, such that the differences between the user's preferences can be utilized positively. The system can generate images based on one user's preferences and thus be considered as novelty by another user. In this scenario, the second user has no involvement in the system. However, there is no implemented fitness function that is able to cover every user's preferences.

The experiments shown in Figure 2 confirm that the parameters used for the fitness function influence the style of the results, as noted by den Heijer & Eiben (2010). They pointed out that this might not be beneficial for the application. However, our analysis shows that for some use cases it might be beneficial, e.g., for "Dynamic Ambient Decoration" and "Therapeutic Art", while other use cases might require more novelty and artistic freedom, e.g.,"Artist's Work Tool" and "AI Art Generator". where the computer should be able to generate high quality and novel art, either through interplay with and guidance from an artist or completely self-sustained.

The framework enables improvements in several directions. For instance, evolutionary algorithm improvements (different genotype, phenotype, mutation and fitness function solutions), alternative approaches to music-to-image transformation, utilizing additional music and image features to enrich the results, interfacing other music and image systems and platforms and using the additional information and knowledge they can offer, and interacting with the end-user in new ways, e.g., creating web platforms that can learn by interaction with user groups (inspired by Trujillo et al., 2013 and García-Valdez et al., 2013).

References

Ashmore, L. & Miller, J., (2004). Evolutionary Art with Cartesian Genetic Programming. *Technical Online Report*.

Baluja, S., Pomerleau, D. & Jochem, T., (1994). Towards automated artificial evolution for computer-generated images. *Connection Science*, 6:325–354.

Bredesen, T.H., Carlsen, J., Seem Koren, H.K., Serussi, S. & Strømjordet, E. (2016). *Sågawave, Creating visualizations based on attributes in music*. Trondheim: NTNU.

Clune, J. & Lipson, H., 2011. Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In *ECAL*., 2011.

Cohen, H., 1988. How to Draw Three People in a Botanical Garden. In *AAAI*., 1988.

Colton, S., (2012). The Painting Fool: Stories from Building an Automated Painter. In J. McCormack & M. d'Inverno, eds. *Computers and Creativity*. New York: Springer. pp.3–36.

den Heijer, E. & Eiben, A.E., (2010). Comparing Aesthetic Measures for Evolutionary Art. In *Applications of Evolutionary Computation: EvoApplications 2010, Proceedings, Part II*. Springer. pp.311–320.

Dumoulin, V., Shlens, J. & Kudlur, M., (2016). A Learned Representation For Artistic Style. *arXiv*, Available at: https://arxiv.org/abs/1610.07629.

Eiben, A.E. & Smith, J.E., (2015). Interactive evolutionary algorithms. In *Introduction to Evolutionary Computing*. Berlin Heidelberg: Springer. pp.215–222.

Galanter, P., (2010). The Problem with Evolutionary Art Is. In *Applications of Evolutionary Computation: EvoApplications 2010, Proceedings, Part II*. Springer. pp.321–330.

Galvin-López, E., McDermott, J., O'Neill, M. & Brabazon, A., (2011). Defining locality as a problem difficulty measure in genetic programming. *Genetic Programming and Evolvable Machines*, 12:365–401.

García-Valdez, M., Trujillo, L., de Vega, F.F., Guervós, J.J.M., & Olague, G. 2013. Evospace-interactive: A framework to develop distributed collaborative-interactive evolutionary algorithms for artistic design. In *International Conference on Evolutionary and Biologically Inspired Music and Art*., 2013.

Johnson, C.G., (2016). Fitness in evolutionary art and music: a taxonomy and future prospects. *International Journal of Arts and Technology*, 9:4–25.

Klinger, A. & Salingaros, N.A., (2000). A pattern measure. *Environment and Planning B: Planning and Design*, 27:537–47.

Kowaliw, T., Dorin, A. & McCormack, J., 2009. An empirical exploration of a definition of creative novelty for generative art. In *Australian Conference on Artificial Life*, 2009.

Lehman, J. & Stanley, K.O., 2010. Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*., 2010.

Lewis, M., (2008). Evolutionary visual art and design. In *The art of artificial evolution*. Springer. pp.3–37.

Machado, P. & Cardoso, A., 1998. Computing aesthetics. In *Brazilian Symposium on Artificial Intelligence*., 1998.

Machado, P. & Cardoso, A., (2002). *All the truth about NEvAr*. Portugal: Applied Intelligence 16(2):101–119

Re, A., Castelli, M. & Vanneschi, L., 2016. A Comparison Between Representations for Evolving Images. In *International Conference on Evolutionary and Biologically Inspired Music and Art*., 2016.

Secretan, J., Beato, N., D'Ambrosio, D.B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J.T., & Stanley, K.O. (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19:373–403.

Sims, K., (1991). *Artificial evolution for computer graphics*. ACM.

Todd, S. & Latham, W., (1992). *Evolutionary art and computers*. London: Academic Press, Inc.

Trujillo, L., García-Valdez, M., Fernández-de-Vega, F. & Merelo, J.-J., 2013. Fireworks: Evolutionary art project based on evospace-interactive. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*., 2013.

Vinhas, A., Assunção, F., Correia, J., Ekárt, A., & Machado, P., 2016. Fitness and novelty in evolutionary art. In *International Conference on Evolutionary and Biologically Inspired Music and Art*, 2016.

Woolley, B.G. & Stanley, K.O., 2011. On the deleterious effects of a priori objectives on evolution and representation. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*., 2011.