# From Music to Animations: A Computational Creativity Approach

*Isabel Rego Raposo*
isabel.raposo@tecnico.ulisboa.pt
Instituto Superior Técnico, Lisboa, Portugal
October 2021

We present a system that creates two-dimensional abstract animations based on audio files, using a Computational Creativity approach. We start by segmenting the audio, such that each audio section generates its own animation. The animations are created from audio-dependent mathematical functions, obtained by randomly assembling elementary functions and variables into an expression tree. We developed three animation analysis methods to select the final colormaps and functions from a pool of randomly generated ones. Mood Matching looks for audio-animation matches by evaluating their mood. The Preference Model emulates the aesthetic preference of the user, for which we obtained a validation precision of 84.6%. Manual Selection corresponds to a co-creation method, where the user chooses the final colormaps and functions from a set of high fitness samples. The methods were evaluated through online forms, revealing that participants deemed the videos creative, and the visuals were considered to align well with the audio. The Preference Model outperformed Mood Matching, and Manual Selection exhibited potential for its use as a co-creation tool to generate music visualizations. We believe our results further motivate the exploration of Computational Creativity systems based on human preference modeling.
**Keywords:** Computational creativity; Music visualization; Computational aesthetic measures; Human preference modeling.

## I. INTRODUCTION

The technological advancements of modern society have allowed us to listen to songs on demand, and provide a variety of mediums that enhance the auditory experience with the aid of the other senses. Music videos are short videos that complement a song or album with visuals. A person is usually required to manually assemble a video stream that visually matches the music in question in some regard, since the desired aesthetics can greatly vary depending on the song, and even between producers. When the perceived visual mood matches that of the audio, the emotional impact of both the audio and the visual experience can be reinforced [1].

In today's personal computer and smartphone era, music visualization tools are widespread across platforms, and can range from a set of pre-existing visuals to visualizations generated by the combination of certain effects. The goal of music visualization is to display animated imagery that changes according to the audio's progression, and should be highly correlated to music features over time - the audio amplitude and frequency spectrum are usually used. Typical music visualizations created by such tools are distinct from conventional music videos, as they are often real-time generated, and some tools can even display distinct visuals every time the program runs. Compared to music video creation tools, music visualization systems lack flexibility in terms of user manipulation of outputs, and are less likely to generate visuals that adequately represent high-level audio features.

A considerable amount of music video generation tools segment the audio and select video clips from available footage to display in each section, such as Foote et al. [2]. A popular approach to computer-generated visual media was initially described by Sims [3], which introduced the use of symbolic expressions to create original visuals, and has inspired similar works such as [4–6]. Our work was further informed by aesthetic measures used to analyze images and animations, such as the ones presented by Li et al. [7] and by Unemi [6].

This work explores possible solutions to the problem of animation generation from audio through a Computational Creativity (CC) approach. Our work had the following objectives: the visual features should be aligned with the audio features in some manner; the animations should be considered creative; the animations should be enjoyable. Our system creates one animation per audio section by parsing mathematical expressions that depend on the frequency intensities of the audio and its estimated tempo. We developed solutions based on the audio mood and the aesthetic preference of the user, and prototyped a co-creation tool using the previous methods. Videos created in the context of this work can be found in `https://vimeo.com/frommusic2animations`.

We now briefly describe the organization of the sections that follow. We introduce our approach in Section II. Audio analysis is described in Section III, Section IV details our approach to animation generation, and Section V specifies the different mechanisms developed for animation analysis. The evaluation method and the obtained results can be found in Section VI. We conclude by highlighting our main conclusions in Section VII.

## II. APPROACH

We now present our proposed solution to the challenge of creating animations that use audio as the starting point. Figure 1 presents an overview of the system's architecture, indicating the main modules that constitute the system, and the elements that are passed between components. The code was developed in Python.

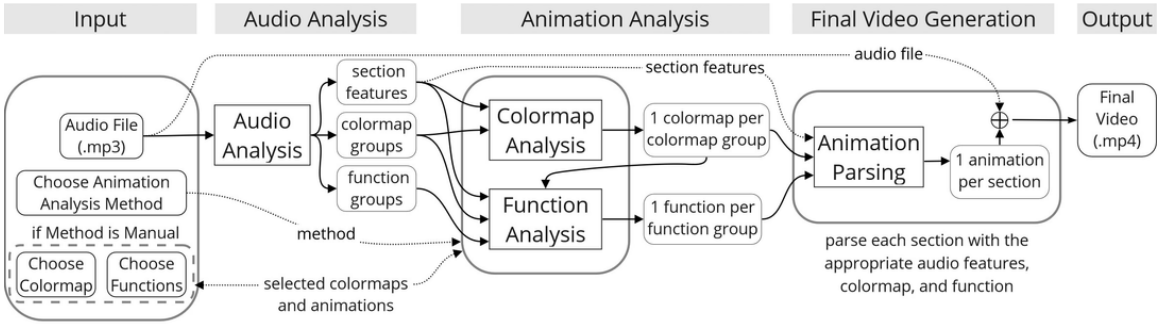



FIG. 1: Overview of the architecture of the system. The main modules that constitute the system are: Audio Analysis, Animation Analysis - which encompasses Colormap Analysis and Function Analysis, and Final Video Generation.

Our approach takes an audio file as input and outputs a video file, which presents abstract animations that were created based on the audio features, accompanied by the original audio. The method used to create animations was based on the seminal work of Sims [3]. We build upon Sims's idea of generating textures using symbolic expressions as the starting point. A frame is achieved by representing a function, and each function is obtained through a random assortment of unit functions which take as arguments the location of the pixel and time-dependent variables. In our case, some of the variables that change according to the frame are audio features or otherwise determined by them, so the animation can be directly influenced by the audio itself. In this way, in each frame, each pixel is mapped to a number, and a colormap is used to decide what color that value corresponds to.

To make the final video more compelling, the audio is sectioned by determining peaks in audio novelty. These sections produce one animation each, which can be generated using unique functions and colormaps, or repeated ones. To reflect the structure of the audio in the video, audio sections that exhibit a similar mood use the same colormap, and audio sections with a similar frequency distribution can use the same animation function. The computations of colormap groups and function groups occur independently.

To determine a colormap to be used, first a set of 250 random colormaps is gathered, and then a selection method is deployed to pick one. The same applies to functions, where one is selected out of a batch of 100 new ones. We designed several methods that can be used to select colormaps and functions: Mood Matching, Preference Model, and Manual Selection. Each method is explained in more detail in Section V.

Once the colormaps and animation functions are determined for every section, it is time to parse all the section animations into the high-resolution frames that will be used in the final video. The audio sections are processed sequentially and, as the frames are computed, they are assembled into the final sequence. Once the audio is added, the final video is complete. Figure 2 shows frames from videos produced by the system.

The following sections detail the intermediary steps needed to compute the video, according to our approach.

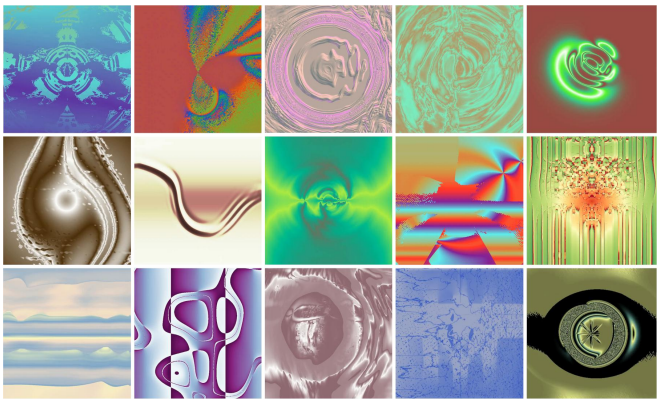


FIG. 2: Frames extracted from videos produced by the system, manually chosen for their appeal and diversity.

## III. AUDIO ANALYSIS

Our system starts with the input of an audio file for which a related video is expected to be generated. In this section, we detail the audio features that are extracted from the file to be further used by future modules.

### A. Audio Features Extraction

To make sure that all audios are uniform, the audio is resampled to have the sampling rate $\omega_0 = 16$ kHz, and analyzed through frames of 0.08 seconds, without overlap. These audio frames will each correspond to a frame in the animation, resulting in a video of 12.5 frames per second (fps).

The frequency intensities are obtained by applying a Fast Fourier Transform (FFT). Following the method described by Lu et al. [8], the intensities are grouped in a way that is similar to the octave scale, forming the Sub-band Intensity features, according to the intervals

$$\left[0, \frac{\omega_0}{2^{n_{subs}}}\right), \left[\frac{\omega_0}{2^{n_{subs}}}, \frac{\omega_0}{2^{n_{subs}-1}}\right), \ldots, \left[\frac{\omega_0}{2^2}, \frac{\omega_0}{2^1}\right],$$

where $\omega_0 = 16$ kHz is the sampling rate and $n_{subs} = 7$ is the number of subbands. This subband division is used throughout the audio analysis and animation generation.

A brief overview of the audio features is represented in Table I, and a more in-depth description of each one can be found in [2] regarding most features, with the exceptions of the MFCC features which were adapted from [8], and the STFT features from [9]. The presented list corresponds to what we call the primary features, as they precede audio segmentation and mood estimation, which are performed in later modules.

TABLE I: Primary audio features, adapted from [2], with the exceptions of the MFCC features which were adapted from [8], and the STFT features from [9]. The *use* column indicates where the features are used: Section Analysis (S), Mood Model (M), Animation Generation (G), or Preference Model (P).

| Feature | Description | use |
|---|---|---|
| Intensity | Spectrum sum of the signal | M,P |
| Subband Intensity | Spectrum sum of the signal over each subband | G,P |
| Subband Intensity Ratio | Spectrum distribution in each subband | M,P |
| Brightness | Centroid of short-time Fourier amplitude spectrum | M,P |
| Bandwidth | Weighted average of the differences between the spectral components and the centroid | M,P |
| Roll off | 95th percentile of the spectral distribution | M,P |
| Spectral Flux | 2-Norm distance of the frame-to-frame spectral amplitude difference | M,P |
| Subband Peak | Average of a percent of the largest amplitude values in the spectrum of each subband | M,P |
| Subband Valley | Average of a percent of the lowest amplitude values in the spectrum of each subband | M,P |
| Subband Contrast | The difference between the Peak and Valley in each subband | M,P |
| Rhythm Strength | The average onset strength in the onset sequence | M,P |
| Average Correlation Peak | The average strength (amplitude) of the local peaks in the auto-correlation curve | M,P |
| avr(A)/avr(V) | The ratio between the average peak strength and average valley strength | M,P |
| Average Tempo | Represents the average speed of the music performance, also referred to as *av_tempo* | S,M, G,P |
| Average Onset Frequency | The ratio between the number of onsets and the corresponding time duration | M,P |
| MFCC | FFT coefficients grouped into 20 bins according to the Mel-frequency scaling | P |
| STFT | Logarithm of the magnitude of STFT coefficients, grouped into 30 bins | S |

### B. Section Analysis

When it comes to music videos, it is frequent to see drastic visual changes matching the beat. Based on the work of Foote et al. [2], we detect temporal peaks in audio novelty, which determine a transition in time from one animation to another. The feature vector of frame $i$ is denoted as $\boldsymbol{v_i}$, and corresponds to the STFT features described in Table I. Cosine similarity is used to compare two frames, defining the Frame Similarity, $S(i, j)$. By computing the similarities between all the frames, we can represent the Self-Similarity of the audio with a 2-dimensional matrix, $S$. Figure 3 shows a detail of the $S$ matrix for the song Prelude In E Minor (opus 28, nº 4) by Chopin.

To approximate the perception of a significant change in the audio, we correlate a checkerboard-like kernel along the diagonal of the $S$ matrix. In our case, the kernel indexes vary from $-L$ to $L$, where $L$ is dependent on the Average Tempo of the audio. Using a Gaussian function to reduce the importance of frames further away from the center, the final kernel corresponds to

$$C(m,n) = \begin{cases} \exp\left(-\frac{m^2+n^2}{2L^2}\right), & \text{for } 0 < m,n \vee m,n < 0, \\ -\exp\left(-\frac{m^2+n^2}{2L^2}\right), & \text{otherwise,} \end{cases} \tag{1}$$

where $m, n \in \mathbb{Z} : m, n \in [-L, L]$ are the kernel indexes. The Novelty Score is formalized as:

$$N(i) = \sum_{m=-L}^{L} \sum_{n=-L}^{L} C(m,n) S(i+m, i+n), \tag{2}$$

where $i$ is the index of the frame of interest. The final step is to compute the local peaks of the Novelty Score. We defined the minimum frame distance that the selected peaks should have as $min\_peak\_distance = int(min(\{0.9 \cdot av\_tempo, 2\})/spf)$ frames. The resulting peaks determine the starting frames of new animations. Figure 3 shows a detail of the Novelty Score and transition points for the song Prelude In E Minor (opus 28, nº 4) by Chopin. The image illustrates that, the more evident the checkerboard-like appearance of the diagonal of the $S$ matrix, the higher the Novelty Score.

### C. Mood Model

A model was trained based on the features used in [8] to estimate the mood present in an audio segment. Following Thayer's model of mood [10], the mood is characterized by two quantities: valence, a metric of how pleasant a mood is; and arousal, which is proportional to how intense or energetic a mood is. These can range from 0 to 1. We use two regression models, one for arousal and one for valence. The DEAM dataset [11] was used to gather song samples with arousal and valence annotations per second. Out of the regression models we tried, the best results were obtained from Support Vector Regression (SVR) models for both arousal ($RMSE = 0.129$, $PCC = 0.890$) and valence ($RMSE = 0.145$, $PCC = 0.819$), which lead to the usage of these models when computing the mood metrics.

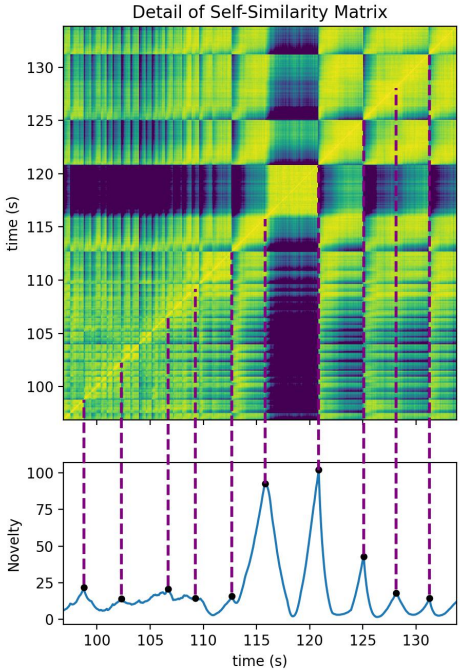



FIG. 3: Close-up of the Self-Similarity Matrix and Novelty Score for Prelude In E Minor (opus 28, n$^{o}$ 4) by Chopin, showing the alignment between the Novelty peaks and the corresponding matrix points.

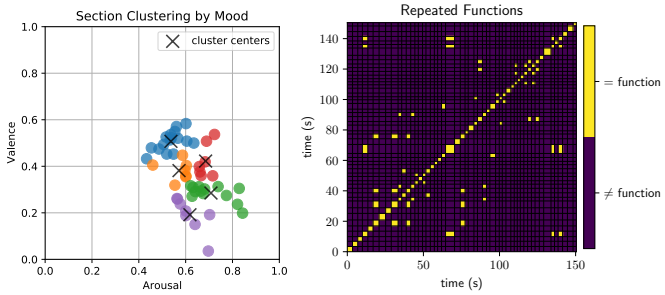
### D. Section Groups

Because we want to use the same colormap for sections with similar moods, the sections are clustered in a hierarchical manner using arousal and valence as the features. The colormap groups are obtained by applying the OPTICS algorithm [12] as exemplified in Figure 4a, such that sections in the same colormap group use the same colors.

We also group sections based on frequency distribution similarity, to form what we call function groups. A section is represented by the average feature vector of all the frames belonging to that section. Each section is compared with the previous sections through cosine similarity, and the one with the highest similarity is considered. If they have a similarity above the threshold (empirically set to 0.9997), then the two are considered a match, and will use the same animation function. Additionally, a section is considered to not have a match if its best match is the one right before it. In this way, we iteratively form groups of matched sections, which dictate the use of the same animation function (Figure 4b).

## IV. ANIMATION GENERATION

The recipes that generate animations have two main components: a colormap and an animation function. We now detail how new colormaps and functions are created.




(a) Colormap Groups (b) Function Groups

FIG. 4: Audio analysis of the song “plyPhon" by Autechre. (a) Section clustering by mood. (b) Indication of the use of the same animation function.

### A. Colormap Generation

Colormaps are functions that map values in $\mathbb{R}$ to the Red, Green, Blue (RGB) space. In our approach, the colormaps only map values from the [0,1] range, to guarantee uniform colormap usage. When fetching a new colormap, after deciding on the number of unique colors, $n_{colors} \in \{2, 3, 4\}$ (proportional to the audio arousal), random values between 0 and 1 are gathered to form a matrix with the shape $n_{colors} \times 3$, where the columns are quantities of Red, Green, and Blue. The color sequence is repeated or mirrored to increase the variability of colors, and there is a chance that some colors are changed to black or white, and all colors can be made lighter or darker. Once we have the final sequence of RGB values, the colors are evenly associated with values in the [0,1] range, and intermediary values are computed by linearly interpolating each RGB value between the two closest colors. Examples of colormaps generated using this method are shown in Figure 5.

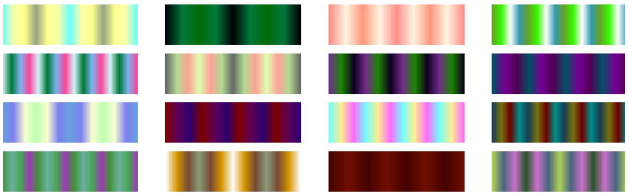


FIG. 5: Examples of randomly generated colormaps.

### B. Function Generation

We now turn our attention to the construction of a function that takes each pixel position and frame index and returns a value in the [0,1] range.

We achieve this by organizing in a tree-like format a few nodes that represent either functions, variables or constants. The final result constitutes the expression tree that determines what is drawn in each frame. The root node outputs values in the [0,1] range that are then

mapped into colors. Since this node is a function with $N$ arguments, then it must have $N$ children nodes, which can be functions, variables or constants, and so on. The terminals (leaves) of the tree must be variables or constants, and the non-terminal nodes (interior nodes) must be functions.

The function is parsed starting from the terminal nodes, then their parent nodes are computed, until the root is reached. To make sure that we have a consistent mapping between any output and a color from the colormap, we perform a transformation of the output of the root into the [0,1] space as follows:

$$output' = \frac{1}{2} \cdot \frac{output}{1 + |output|} + \frac{1}{2}. \tag{3}$$

Figure 6 shows examples of trees and the frames obtained from parsing them.

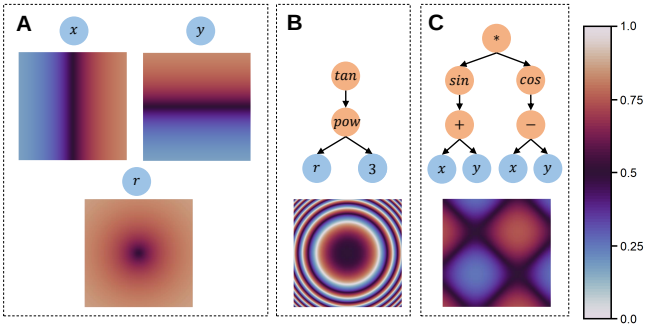



FIG. 6: Examples of function trees and the resulting frames, obtained using the twilight colormap. All the values were transformed through Equation 3. (A) Variables based on pixel coordinates. (B,C) Examples of function trees.

The possible terminal nodes are: constants; variables related to the location of the pixel, $x$, $y$, and $r = \sqrt{x^2 + y^2}$; variables that change with time can be directly obtained from the Subband Intensities, or can be sinusoidal functions with a periodicity of 0.5, 1, or 2 times the Average Tempo; a linear function of time; and one of the pattern nodes. The patterns are treated as terminals because they are computed when the tree is assembled and are frame-independent. These can be: 2D Perlin Noise; angle and radius dependent patterns such as spirals and flower-like textures; and patterns generated by performing multiple convolutions with a random kernel over a pure noise canvas.

The non-terminal nodes represent a function randomly chosen from a function set, and require one or more arguments, up to four. There are multiple possible types of non-terminal nodes. Standard functions corresponds to a set of functions that treat each pixel location and frame index independently. Other non-terminal nodes include: a convolution with a random kernel performed several times over its child node; functions that treat its two child nodes as $x$ and $y$ and create spirals and flower-like textures (based on the pattern described above); and the diffusion node, inspired by reaction-diffusion systems, where the new value in a pixel location at a given frame is dependent on the previous frame and on the surrounding pixels.

The set of standard functions includes: arithmetic operations, trigonometric functions, module, exponential, logarithm, squared, square root, power, Perlin noise (1D, 2D, and 3D), blur, Prewitt, distance, minimum/maximum between two child nodes, smoothclamp, lerp, warp, and the operation **if** $child_1 > child_2$**, then** $child_3$**, otherwise** $child_4$.

For the time-dependent nodes of the function tree, the system fetches the required features (i.e., the Subband Intensities) that match the timestamp of the frame. Figure 7 shows examples of frame sequences obtained from time-dependent expressions.

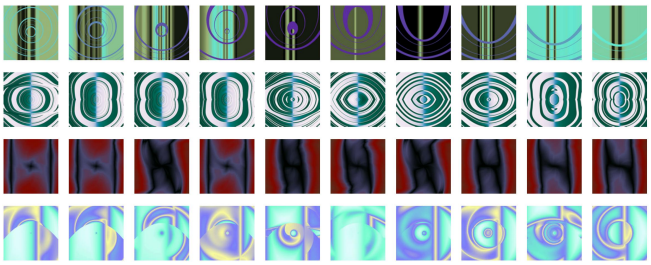


FIG. 7: Examples of animations resulting from our system. Each row shows one frame sequence, and time evolves from left to right.

### C. Animation Features Extraction

For analyzing the animations, we extract the features presented in [7], since the features they considered were shown to be useful to predict the user's aesthetic preference regarding images. We also evaluate how much the colors change between frames, on average, following [6]. We transform information about frames into animation features by computing their average over the animation. All the features are computed on frames with $50 \times 50$ resolution, independently of what the final video resolution will be, to ensure consistency of results. An overview of the extracted features is shown in Table II.

Function Filtering is the enforcement of constraints on the functions generated, to avoid: a large number of frames having similar colors for all pixels, animations perceived as almost static or that look like pure noise, and functions that take too long to parse.

## V. ANIMATION ANALYSIS

In this section, we consider the animation analysis methods individually, detailing how each one selects the colormaps and animation functions.

TABLE II: Animation features that are extracted by the system. The first six features were adapted from [7], and Temporal Variation was adapted from [6]. The *use* column indicates where the features are used: Function Filtering (F), Mood Matching (M′), or the Preference Model (P).

| Feature | Description | use |
|---|---|---|
| Color moments | Three central moments of hue, saturation and lightness | P |
| Lightness feature | Lightness channel based on Benford's Law | P |
| Texture feature | Local binary patterns (LBP) in four ranges | P |
| Image complexity | Information entropy of HSL, RGB and Y709 | P |
| Image order | Low complexity based on fractal compressor | P |
| MC metric | The image complexity (IC) and processing complexity (PC) ratio | P |
| Local Variation | Mean and standard deviation over the animation of $loc\_var$, which is computed as $Prewitt(Grayscale(frame))$ | F, M′, P |
| Local Variation Percentage | Percentage of frames in the animation for which the condition $0.05 < loc\_var$ applies | F |
| Temp. Variation | Mean and standard deviation of $temp\_var$, which is the average distance in the RGB space between pixel colors of consecutive frames in the same positions | F, M′, P |
| Temp. Variation Percentage | Percentage of frame transitions where $0.01 < temp\_var$ | F |
| Movement Correlation | Quantification of correlation between $temp\_var$ and each Subband Intensity | P |

### A. Mood Matching

In Mood Matching, we focus on the arousal and valence of the audio to analyze animations.

#### 1. Colormap Selection

To capture the arousal of the audio in the colormap, we associated the use of more unique colors to higher arousal: for audio arousal below 1/3, $n_{colors} = 2$; for audio arousal above 2/3, $n_{colors} = 4$; and intermediary arousal leads to $n_{colors} = 3$. The variable $n_{colors}$ is provided to the colormap generator, which returns a set of colormap candidates, that are analyzed by valence. To capture the valence of a color, we looked for a function that would provide higher valence for brighter colors and lower valence for darker colors, but at the same time provide a low valence for gray colors and higher for more saturated ones. The function that we believe best captured this idea is:

$$Val = Lig^{10} + 1 - sigm((1 - Sat)^2 + (1 - Lig)^2), \quad (4)$$

where $Sat, Lig \in [0, 1]$ represent the Saturation and Lightness values regarding the HSL color space, and $sigm(\cdot)$ is the sigmoid function. $Val$ is further restricted to the [0,1] domain. The final valence of a colormap corresponds to its average color valence, computed through the discretization of the colormap into 50 colors. Example colormaps and their valence are presented in Figure 8. The Mood Matching method chooses the colormap with the valence closest to the audio valence.

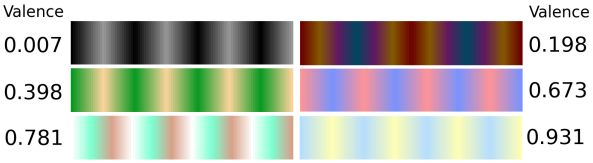



FIG. 8: Valence values for different colormaps.

#### 2. Animation Selection

To compare each animation with the audio mood, we now only consider the arousal of the audio. It is our understanding that, for higher arousal, an animation should have more edges and abundant movement, while lower arousal is better represented by frames with fewer edges and slower movements. For every sample animation, the $arousal\_distance$ is defined as the maximum between $|Audio\_Arousal - mean\_Local\_Variation|$ and $|Audio\_Arousal - mean\_Temporal\_Variation|$. Because we want to guarantee that this value is small, the animation with the lowest $arousal\_distance$ out of the sample set is the one that is selected.

### B. Preference Model

We developed a method that takes the user's aesthetic preference into account when looking for audio-animation matches. A dataset was developed to gather user classifications of animations given a set of audios, and a selection process was created to train and select the model with the highest precision. The final model estimates the fitness of each animation based on the audio, and the one with the highest fitness is selected.

#### 1. Training and Model Selection

Using the DEAM dataset, 55 audio samples of 10 seconds were gathered, such that the final set of songs would include a minimum of 4 examples for each music genre. For each audio sample, 15 different animation samples were generated independently, each one with a random colormap and function. The short videos of $100 \times 100$ pixel resolution are shown to the user through an interface, where they are animated synchronously with the audio. For every audio, the user indicates for its 15 animations which ones they prefer (class +1), which ones they feel neutral about (class 0), and which ones they do not like to see with that audio (class -1). The same process is repeated for the next audio, and so on. Figure

9 illustrates the interface used to collect user classifications.

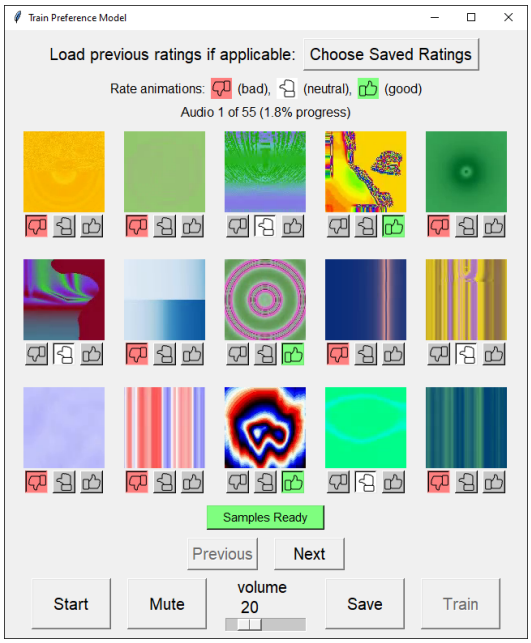



FIG. 9: Preference Model interface used to classify each audio-animation pair.

An audio-animation pair is represented by concatenating the audio and animation features indicated in Tables I and II with "P" in the column *use*. The final classifications are used to train several models, namely Linear Regression, Decision Tree, Support Vector Regression, and Multilayer Perceptron. The model and parameters with the highest average precision were selected, corresponding to an SVR model, obtaining a validation precision of 84.6% regarding the correct classification of class +1 (regression value equal to 0.5 or above). The model was trained using our classifications, and the model selection process can result in a different model for classifications provided by another user.

#### 2. *Colormap Selection*

In the context of the Preference Model approach, selecting a colormap means choosing a colormap that increases the likelihood of generating animations with higher fitness. Three random functions are created, and three sections from the colormap group are chosen. The Mood Matching method is used to get 5 new colormaps, and we use each one to create several sample animations - one animation per colormap-function-section combination. The fitness of every animation is computed using the preference regression model, and the colormap that resulted in the highest average fitness is selected.

#### 3. *Animation Selection*

The model trained on the user's preference is used to compute the fitness of every animation from a randomly acquired set generated using the same audio, and the function that generates the animation with the highest fitness is selected for the function group.

### C. Manual Selection

Besides the autonomous methods, we developed a solution that allows the user to choose the samples that will be used in the final video. This approach uses the automatic methods to filter some of the samples so that only the top ones are shown, and then the user can choose their favorite. This approach can be seen as an example of co-creation.

#### 1. *Colormap Selection*

The process begins with the generation of 250 colormap samples for the colormap group, from which the top 5 according to Mood Matching is selected to be shown to the user. The user has the possibility of asking for 5 more colormaps selected from a new set of 250 examples, until they are pleased with one of them. Once the user has found a colormap they are pleased with, they indicate which one they want, and then the process is repeated for the next colormap group.

#### 2. *Animation Selection*

For animation selection of each function group, the Preference Model is used to filter the animations. Given a random set of 100 functions, the animations are created based on the first section of the function group. Each fitness is computed, and the top 15 is shown to the user through a user interface, ordered from highest fitness to lowest. All the animations move at the same time, synced with the audio. The user indicates which one should be used in the final video, and the process is repeated for the next function group. This interface is shown in Figure 10.

## VI. EVALUATION

To evaluate our system, we selected five songs, and each one was used to generate one video per animation analysis method: Mood Matching, Preference Model, and Manual Selection. An additional control video was generated per song, by using a video that was created based on a different song and switching the audio. The videos correspond to 12.5 fps and $720 \times 720$ pixels. The songs were chosen for their diversity in

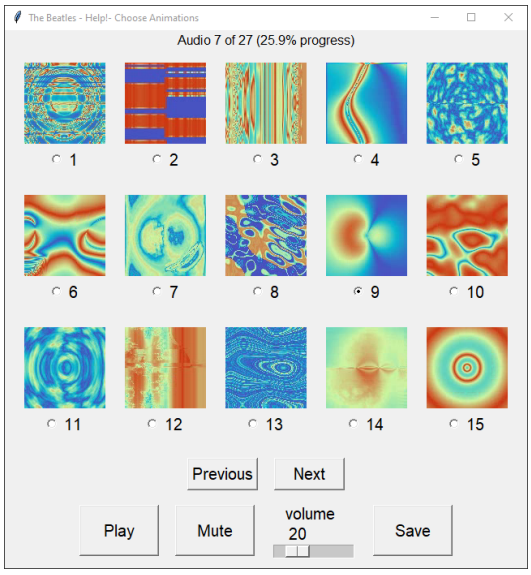



FIG. 10: Manual Selection interface to choose the animations.

TABLE III: Songs used to evaluate our system, and the corresponding forms.

| Form | Music |
|---|---|
| A | Autechre - "plyPhon" |
| B | Chet Baker - "It Could Happen To You" |
| C | Chopin - "Prelude In E Minor (opus 28, n$^{o}$ 4)" |
| D | Coro Madrigale Slovenico - "Sanctus" |
| E | The Beatles - "Help!" |

genre and mood, and are listed in Table III. Five forms were developed such that, for each form, only one song and the videos associated with it were shown. The videos used in this evaluation can be found in `https://vimeo.com/frommusic2animations`. No information was provided to the participants regarding how the animations were created, the videos were referred to using a version number, and the method versions were not the same in each form.

### A. Form description

We now provide a short description of the form structure used. When asking questions about the artifacts, the audio questions were asked first, then the questions for version 1 of the video, then version 2, and so on. After the demographic information was gathered, the participant was shown the audio and videos and was asked to provide adjectives or expressions that describe them. Then, we collected opinions resorting to the Likert scale method [13], by presenting statements to the participant, and then asking them to indicate how much they agree or disagree using the following scale: 1 - Strongly Disagree, 2 - Disagree, 3 - Undecided, 4 - Agree, 5 - Strongly Agree. The Likert scale questions were different for audios and videos. In the case of the audio, the statements were "I like it a lot" and "I consider it very creative". Then, for each video, we presented the statements "I consider it a music video", "I like it a lot", "I consider it very creative", "I consider it abstract", and "I think the visuals go well with the audio". The participant was also asked to choose all the adjectives that apply out of a list. These adjectives were: exciting, calm, happy, enjoyable, boring, surprising, sad, aggressive, funny, interesting, disgusting, fearful, confusing, numb, and tender. The last question asked the participant to order the video versions from most preferred to least preferred.

### B. Results and Discussion

In total, there were 167 form submissions, obtained from 84 different people. Each form had a similar answer count, between 32 and 35.

Concerning the adjectives that participants selected out of a list for the audios and videos, Figure 11 shows the Bhattacharyya distance [14], $D_{BC}$, between the adjective distribution for the audio, and the distributions regarding the videos.

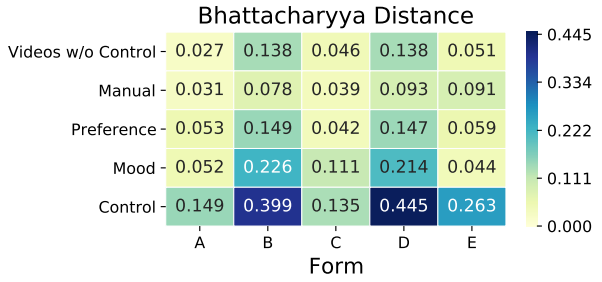



FIG. 11: $D_{BC}$ between the adjective selection distributions of audio and the animation methods for each form.

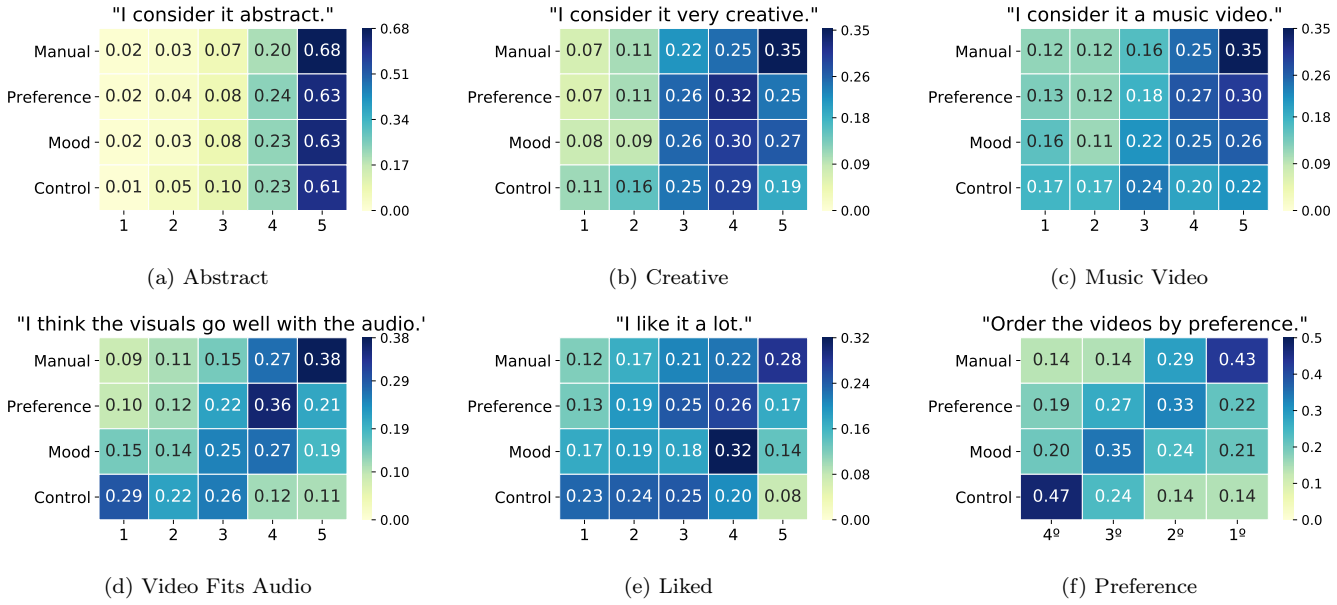
With a few exceptions, the manual method distributions obtained better matches with the audio, followed by the preference method, the mood method, and finally the control method. In addition, some songs generated smaller distance metrics than others. In particular, the songs It Could Happen To You and Sanctus had videos for which the selected adjectives were less related to the audio when compared to the other songs. Overall, the Bhattacharyya distances for the distributions obtained by considering all video versions except control (Videos w/o Control) were always less than 35% of their control counterparts, which indicates a correlation between the audio perception and the visual perception that these animation methods produce.

Table IV shows metrics for the Likert scale questions, and Figure 12 shows answer distributions regarding each option of the Likert scale questions and preference order.

Overall, the videos were overwhelmingly considered abstract, even the control videos, as contemplated in Figure 12a. Regarding the answer analysis presented in Figure

TABLE IV: Metrics per question and per method: mode (Mod), median (Med), average (Avg), and standard deviation (Std).

| | Abstract | | | | Creative | | | | Music Video | | | | Video Fits Audio | | | | Liked | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Mod | Med | Avg | Std | Mod | Med | Avg | Std | Mod | Med | Avg | Std | Mod | Med | Avg | Std | Mod | Med | Avg | Std |
| Manual | 5 | 5 | 4.49 | 0.89 | 5 | 4 | 3.71 | 1.25 | 5 | 4 | 3.59 | 1.38 | 5 | 4 | 3.73 | 1.31 | 5 | 4 | 3.38 | 1.36 |
| Preference | 5 | 5 | 4.43 | 0.91 | 4 | 4 | 3.57 | 1.16 | 5 | 4 | 3.49 | 1.37 | 4 | 4 | 3.47 | 1.22 | 4 | 3 | 3.17 | 1.28 |
| Mood | 5 | 5 | 4.44 | 0.90 | 4 | 4 | 3.58 | 1.21 | 5 | 4 | 3.35 | 1.39 | 4 | 3 | 3.20 | 1.31 | 4 | 3 | 3.07 | 1.33 |
| Control | 5 | 5 | 4.37 | 0.95 | 4 | 3 | 3.29 | 1.26 | 3 | 3 | 3.11 | 1.39 | 1 | 2 | 2.54 | 1.33 | 3 | 3 | 2.65 | 1.26 |




(a) Abstract (b) Creative (c) Music Video

(d) Video Fits Audio (e) Liked (f) Preference

FIG. 12: (a)-(e) Answer distributions regarding the Likert scale questions. The values represent: 1 - Strongly Disagree, 2 - Disagree, 3 - Undecided, 4 - Agree, 5 - Strongly Agree. (f) The distributions for the order of preference of the videos.

12b, while the manual method seems to be the most effective at producing videos deemed creative, with a mode of 5, the preference and mood methods seem to perform relatively well, with classifications mostly ranging from 3 to 5, with 4 as the median and mode. The control videos also presented a considerable performance in this regard, which we believe shows our system's ability to generate creative animations, even when they don't match the audio.

Regarding the statement that the videos were music videos (Figure 12c), the control video produced opinions almost evenly distributed, then the distributions get more skewed to agreeing with the statement as the methods go from mood, to preference, to manual. This is telling of the abilities of the different methods for selecting the best animations to go with an audio.

From the analysis of Figure 12d, the manual videos appear to be ones with visuals that were considered to go well with the audio more often. In addition, the automatic methods also performed considerably well, with the preference method producing better results.

Opinions greatly diverged regarding liking the videos, as can be seen in Figure 12e, which is coherent with the notion that aesthetic tastes are based on subjective personal criteria, at least to some extent. Nevertheless, in general, the control videos were not strongly liked, and the manual videos, on the other hand, were the most likable of all.

Inspecting Figure 12f, we can see that the manual videos were often the favorite ones, followed by the preference videos, then the mood videos, and the control videos were frequently placed last. This is consistent with the analysis regarding the analysis of whether the videos were considered music videos, whether the visuals go well with the audio, and if the videos were liked.

Regarding our objectives, we believe our solutions showed positive results, as the automatic methods were able to surpass the control versions in terms of perception of audio and visual alignment. Furthermore, the Manual Selection results reveal its potential as a co-creation tool.

## VII. CONCLUDING REMARKS

We have presented a system that creates original abstract videos with animations that depict a song. The driving idea behind our system was the generation of animations by drawing time-dependent mathe-

matical expressions frame by frame, which should move in accordance with the audio. The videos generated in the context of this work can be consulted in `https://vimeo.com/frommusic2animations`.

The audio is segmented according to audio novelty and its average tempo, such that each audio section generates its own animation. Audio sections can use a repeated colormap or function according to section similarity. New colormaps are generated by randomly assembling sequences of RGB values. The animation functions are obtained by arranging mathematical operations, constants, pixel location variables, and time-dependent variables. We use the frequency intensities extracted from the audio as the temporal variables, along with sinusoidal functions that align with the average tempo of the audio.

To select the colormaps and functions to be used, first a collection of samples is generated, then the final ones are selected using one of the animation analysis methods.

The first developed method, Mood Matching, is focused on the audio mood for each section. It estimates the average color mood by evaluating color Lightness and Saturation, and estimates the arousal of animations as being proportional to the average amount of detail and color change between frames.

The Preference Model was generated by training an SVR model on audio-animation pairs and user-provided classifications regarding their fitness. The selected model, SVR, obtained a validation set precision of 84.6%.

A final animation analysis method, Manual Selection, relies on a human user to select the best colormaps out of a collection filtered according to Mood Matching, and the best animation functions out of a set filtered using the Preference Model.

Online forms were deployed to measure our system on different metrics, producing positive results. Specifically, evaluation of each method on five songs indicated that the Preference Model performed better than Mood Matching, and the Manual Selection method showed promising results for its use as a co-creation tool to generate music visualizations that fit the audio.

We hope our work inspires further discussion regarding creative artifacts, the creative process, and the possibility of creative computational systems. We believe the results obtained in this work further motivate the study of human preference modeling, with applications not only in the evaluation of aesthetic measures, but also regarding the development and improvement of co-creation tools.

It would be valuable to explore other ways of animating to music, as we have merely explored one possible approach. Our system could be improved through time optimization, which would make viable the deployment of a Genetic Algorithm to iteratively increase the fitness of the animations. An app for video creation could be developed, allowing further study of the presented methods, and other features, models, parameters, and training datasets could be explored.


[1] M. Boltz, "Music videos and visual influences on music perception and appreciation: Should you want your mtv?" *The psychology of music in multimedia*, pp. 217–234, 2013.

[2] J. Foote, M. Cooper, and A. Girgensohn, "Creating music videos using automatic media analysis," in *Proc. of the ACM International Multimedia Conference and Exhibition*, 2002, pp. 553–560.

[3] K. Sims, "Artificial evolution for computer graphics," in *Proc. of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, jul 1991, pp. 319–328.

[4] P. Machado and A. Cardoso, "All the truth about NEvAr," *Applied Intelligence*, vol. 16, no. 2, pp. 101–118, feb 2002.

[5] D. A. Hart, "Toward greater artistic control for interactive evolution of images and animation," in *Workshops on Applications of Evolutionary Computation*, 2007, pp. 527–536.

[6] T. Unemi, "Automated daily production of evolutionary audio visual art - an experimental practice," in *Proc. of the 5th International Conference on Computational Creativity*, 2014.

[7] Y. Li, C. Hu, L. L. Minku, and H. Zuo, "Learning aesthetic judgements in evolutionary art systems," *Genetic Programming and Evolvable Machines*, vol. 14, no. 3, pp. 315–337, 2013.

[8] L. Lu, D. Liu, and H. J. Zhang, "Automatic mood detection and tracking of music audio signals," in *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, jan 2006, pp. 5–18.

[9] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[10] R. E. Thayer, *The biopsychology of mood and arousal*. Oxford University Press, 1990.

[11] A. Aljanaki, Y.-H. Yang, and M. Soleymani, "Developing a benchmark for emotional analysis of music," *PLoS ONE*, vol. 12, no. 3, 2017.

[12] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.

[13] R. Likert, "A technique for the measurement of attitudes," *Archives of Psychology*, 1932.

[14] A. Bhattacharyya, "On a measure of divergence between two multinomial populations," *Sankhyā: the indian journal of statistics*, pp. 401–406, 1946.