FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Physics Based Music Visualization

Subtitle

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Software und Information Engineering

eingereicht von

Juergen Giefing
Matrikelnummer 9805067

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Stefan Ohrhallinger
Mitwirkung: Alexander Hauer
Andreas Schmid

Wien, 18.12.2013

(Unterschrift Verfasser) (Unterschrift Betreuer)

FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Physics Based Music Visualization

Subtitle

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Software and Information Engineering

by

Juergen Giefing
Registration Number 9805067

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Stefan Ohrhallinger
Assistance: Alexander Hauer
Andreas Schmid

Vienna, 18.12.2013

(Signature of Author) (Signature of Advisor)

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

Juergen Giefing
Murlingengasse 46/2/1, 1120 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasser)

Kurzfassung

Historisch gesehen, gibt es eine Reihe von Ansätzen, die Zusammenhänge zwischen Musik und Visuellen Elementen, vor allem Farbe zu erkunden und wissenschaftlich zu analysieren. Zu nennen sind in diesem Zusammenhang beispielsweise die Arbeiten von Arnheim [1] sowie Schloss und Palmer [10] [16] [14] [13]. Palmer hat in seinen Versuchen nachgewiesen, dass die Verbindung zwischen Musik und Farbe, wie schon von Arnheim vermutet, auf der emotionalen Bedeutung für den Betrachter bzw. Zuhörer beruht. Das Ziel dieser Arbeit bestand darin, diesen Zusammenhang weiter zu erforschen, indem auch andere visuelle Parameter wie Bewegung, Form und Größe miteinbezogen werden und auf Basis dieser Erkenntnisse den Prototypen einer möglichen Visualisierungssoftware zu erstellen, die Musik auf Basis eines flexiblen Mappings und psychologischer Erkenntnisse visualisiert. Im ersten Teil unserer Versuche, wurden Testpersonen gebeten, Ausschnite aus Musikstücken sowie kurze tonlose Animationen unbhängig voneinander, aber anhand derselben Kriterien zu bewerten. Aus den Ergebnissen lassen sich deutliche Korrelationen zwischen einzelnen Attributen und der Wahrnehmung durch die Zuhörer/Betrachter erkennen. Im zweiten Versuch wurden die Testpersonen gebeten, zu bewerten, wie stimmig die daraus entstandenen Visualisierungen auf Basis unseres Prototypen von ihnen wahrgenommen werden. Die Annahmen konnten dabei nicht bestätigt werden. Es wird versucht, zu eruieren, woran der zweite Test scheiterte und welche Maßnahmen ergriffen werden können, um den oben beschriebenen Ansatz erfolgreich umzusetzen.

Abstract

Historically, there have been a number of approaches to analyze and explain the relationship between music and visual elements, particularly colors. Arnheim [1] has done important work in that context, as well as Palmer and Schloss [10] [16] [14] [13]. Palmer has shown in his experiments, that music and color are coupled through emotion, like Arnheim had assumed before. The goal of this thesis was to investigate this connection more in detail by considering also other visual parameters like motion, shape or size and to implement a prototype of a visualization application based on the insights gathered during our user studies. This application should be able to visualize music based on a flexible mapping and psychological knowledge. During the first part of our user studies, test persons were asked to rate parts of songs as well as animations without sound independent from each other but using the same rating scales. The results show strong correlations between single attributes and the perception of the test persons. During the second part of the user studies, the test persons were asked to rate the accordance between the songs from the first round and the visualizations created based on the results of the first round. Our assumptions could not be confirmed in that experiment. We try to determine the reasons, why the results of the second round were not as expected and what steps could be taken to refine our approach and implement it in a successful manner.

Contents

CHAPTER 1

Introduction

1.1 Motivation

The last decades in media and arts were strongly characterized by mixed media approaches in various forms. especially the invention of the sound film changed the possibilities in matters of acoustic/visual elements and their interaction.

Almost one and a half decades after The Buggles released their hit “Video killed the radio star”, the music video as most prominent form of fusion between music and moving pictures seems to lose importance whereas more or less spontaneous visualisations in various forms become a more and more relevant art form. Thanks to the advanced technical possibilities, live visuals controlled by VJs have become part and parcel of discotheques and even live concerts. At the same time the technical progress has also offered completely new ways to interpret music being played on a computer in a visual way completely automatically. During the last couple of years these visualisations changed from quite simple geometric forms moving across the screen in different colors to great artworks with amazing effects compared to what we knew a couple of years ago. However, despite the technological progress, especially in the area of pc music players, a felt gap between music and the optical equivalent seems to remain.

1.2 Problem Statement

Surprisingly often, the visualisation generated by most visualisation modules of popular mp3 players looks impressive, but still does not really respond to the track or song it belongs to. Music and visualisation often seem to coexist side by side instead of creating a bigger whole piece of art together.

1.3 Aim Of The Work

Therefore we want to figure out whether there are ways using the human psychology to develop visualisations of music, that maybe will not look better in terms of aesthetics but feel more natural to the recipient. the main thesis of this work is, that it is possible to develop such a kind of visualisation based on feelings or more precisely emotions, humans connect with music as well as optical impressions like color, form or motion.

Based on physical parameters gained from the music file, we want to detect events and fragments charged with emotion. on the other side we have also a connection between emotion and optical events and parameters. Each of these connections is based on psychological knowledge and together it shall allow us to visualise music in a way, that feels more natural to the recipient even despite the fact it will not use the amazing effects we are used to because they are standard in popular visualisation plugins nowadays.

CHAPTER 2

State Of The Art

2.1 Literature Studies

Perception And Emotion

In 1986, Rudolf Arnheim claimed, that there were systematic relationships betweeen color and music and they were mediated by emotion. [1] This thesis was proven 2013 by Stephen Palmer, Karen Schloss, Zoe Xu and Lilia Prado-Leon in their article "Music-color associations are mediated by emotion" [10] based on a user study. They provided a set of 37 colors and a set of 18 brief samples of classical music. The test persons were asked to pick the colors that went best and the ones that went worst for them with every single piece of music. During the second part of the study they had to rate every color and every music sample by terms of emotional associations. As they had expected, they found systematic mappings between the dimensions of color and music. Schloss and Palmer also published a number of related studies, where they illustrated specific interrelations betweeen color and music, like eg. [16]. In this article Schloss shows the correlations between tempo and color. She also considers factors like note density or rhythm in her experiments.

Music

A common way to measure the musical experience of test persons is the semantic differential technique introduced by Charles Osgood [9]. There are a couple of scales available using the semantic differential developed specifically for music. However, there are also other approaches to measure musical experience. Kaarle Nordenstreng for example compares this technique to the technique of similarity ratings [8] and shows, that they are in fact measuring the same thing.

However, because of the relative complex rules in theories of harmony, it is not trivial to automatically categorize music on a higher level and map it to emotional states on that level.

Color, Shape and Motion

As mentioned above, there are numerous articles with the focus on the emotional perception of color. This field of science has a rather long history. Pretty well known are for example the theories of Johannes Itten [7] or Johann Wolfgang von Goethe [17], although parts of the latter one are not undisputed. During the last decades numerous scientists have done important work on the field of the psychological aspects of color perception, like mentioned above.

Particularly interesting for our field of interest is the question for cultural differences in the meaning and therefore perception of color. This even implies possible disparities based on the linguistic background as discussed in [4].

Because we are talking about colors on the screen, we will focus only on the RGB (red, green, blue) color scheme usually used to define colors on screens [15].

Interdependencies and Correlations betweeen visual and acoustic perception

Quite obviously there are numerous parallels between the visual and the acoustic world. Both, colors and tones, are often illustrated as circles, distances between to single items can be calculated and we are often even using the same vocabulary when talking about colors and tones.

The connection of and the similarities between these two worlds have often been the topic of science and arts. In medicine and psychology, synaesthesia is a quite interesting field of research. Wikipedia defines synaesthesia as “a neurological condition in which stimulation of one sensory or cognitive pathway leads to automatic, involuntary experiences in a second sensory or cognitive pathway”. It has been studied and analyzed in numerous ways. The articles of Richard Cytowic [3], David Brang [2] and Gian Beeli, Michaela Esslen and Lutz Jaencke give interesting insights into this topic.

The cultural aspect of this coupling is often referred to as “intermedia”. The term was introduced by Dick Higgins in the nineteen sixties for inter-disciplinary activities of different kind and is explained in detail in his article [6].

2.2 Analysis

The current range of media players convinces with a wide range of visualization plugins creating astonishing visuals on top of songs. Some of those plugins also provide numerous preferences for the user, allowing to tune the results of the visualization. However, the possibilities regarding visualization are very limited on a higher level. It’s usually not foreseen to let the user choose visual actions based on acoustic commands and to even provide those actions with individual parameters.

Andreas Schmid gives a more detailed summary of the currently available range of media players and their plugins as well as their configuration options in his bachelor thesis [11].

CHAPTER 3

Methodology

We use the work of Palmer as a starting point for our own work. As mentioned above, Palmer showed, that music is coupled to color through emotion. Based on that knowledge, we go one step further taking a look not only at the coupling between music and color, but also at the coupling between music and other aspects of visual perception. We assume, that music and the whole range of visual elements are coupled through emotion. Therefore, our thesis is, that developing a mapping of musical to visual elements based on that knowledge should lead to a subjectively more natural visualisation.

3.1 Example Mapping Based On User Studies

In our first series of user studies, we asked our test persons to tell us their emotions when listening to a piece of a song. Those pieces had a length of between 13 and 40 seconds.

The songs used, each of them in a midi version and therefore independent from the lyrics. The songs we used were:

- Cryin - Aearosmith
- Crying at the discotheqe - Alcazar
- Everyday People - Arrested Development
- Eleanor Rigby - The Beatles
- Pick up the pieces - Average White Band
- Rain - Annie Lennox
- Respect - Aretha Franklin
- School’s out- Alice Cooper

- Stayin Alive - Bee Gees
- TNT - AC DC

The sample of test persons consisted of 19 persons of different ages, partly male, partly female and with completely different affinity concerning music and graphics. Some of the test persons, for example were professional musicians, some of them were not really deep into music. The same range of interest and affinity was there concerning the affinity to graphics.

More detailed information on the sample of test persons can be found in the appendix.

The sample of songs was selected based on a couple of parameters we used to categorize the music we gave to our test persons. The parameters were

- Tempo
- Instrumentation
- Mode

Because the tempo, measured in beats per minute and the number of instruments used are numerical values, we were able to use them directly. Concerning the mode, we only differentiated between minor and major mode to keep it simple.

Some more parameters probably worth measuring are mentioned later on in the chapter "Future Work". More information on the sample of songs can be found in the appendix.

Like Palmer and his colleagues, we used the semantic differential technique developed by Charles Osgood to get the emotions in a measurable form. We used the scale of Osgood himself consisting of the three axes activity, potency and evaluation he proved using factor analysis to be the only three completely independent from each other. [9] Each axis allowed a rating from one to ten. One meant, the song was perceived not active, potent or positive at all, ten meant, it was perceived very active, potent and positive.

Furthermore we asked our test persons to rate visual elements the same way. Therefore they watched ten simple animations in random order. Each animation ran in the loop mode and was categorized using the following parameters:

- Color
- Shape
- Motion (rotational)
- Motion (translational)
- Number of items
- Size

We first thought about using discrete colors (e.g. the primary and secondary colors in the RGB color space), but then decided to take a deeper look at it and split up each color into its amount of red, green and blue. That way we wanted to get answers allowing a more detailed analysis of the answers considering the single color channels.

The shapes we used were rather simple primitives:

- Dots
- Lines
- Circles)
- Triangles
- Rectangles

The number of items used in the animation was categorized into

- one
- few (2-5)
- many (more than five)

The size parameter had two discrete values: small and large.

More detailed information on the animations can be found in the appendix. The test persons were interviewed sitting exactly in front of the screen without any acoustic distraction during the interview. We also took care, that no test person knew any answers given by another test person, who had been interviewed before.

Based on the results of those interviews, we developed an exemplary mapping trying to map musical elements to their corresponding visual equivalents according to the average rating in our user studies. To have another mapping we could use to compare with, we also developed another one, the so-called anti-mapping, mapping musical elements to visual elements on a random basis. According to our thesis, this second mapping should be considered as significantly less matching than the first one within our second round of interviews.

3.2 Testing The Mapping Based On User Studies

To validate our hypothesis, we asked our test persons to watch fifty different videos, each combining one piece of music from the first round with the automatically generated visual by one of the following players:

- Windows Media Player with
- WinAmp
- iTunes

- our own prototype using the optimized mapping
- owr own prototype using the contrary mapping

The prototype mentioned above was designed and implemented by Alexander Hauer, Andreas Schmid and myself. The main part of my contribution was the application architecture and the implementation of the mapping component. Alexander Hauer was primarily engaged in designing and implementing the analysis part, whereas Andreas Schmid was responsible for the visualisation part. A rough overview of the whole application is given in the Chapter "Implementation". More detailed information about the analysis and the whole analysis process can be found in Alexander Hauers Bachelor Thesis [5]. Further information about the visualization part, the libraries used for visualization and the concepts behind our exemplary theme (see chapter "Implementation") can be found in the Bachelor Thesis of Andreas Schmid [11].

Each test person had to say how good each of the visualizations corresponded to the music he or she listened to on a scale from one to five, where one was meant to be "not corresponding at all" and five meant "perfectly corresponding". To avoid misunderstandings, we also added another question. The test persons also had to say how much they liked the visualization in terms of aesthetics. This way we wanted to avoid, that the test persons mixed up those two aspects. To keep thing simple and have a common wording, we decided to name the first one "accordance" and the other one "aesthetics".

Using this second part of our user study, we wanted to verify, that the example mapping was rated significantly better than the anti-mapping and even better than those of the popular media players.

Contrary to the first round, we had to save time. Therefore we asked our test persons to do answer these questions online, watching the video in the browser. They were asked to watch the videos alone, without any noisy background and sitting exactly in front of the screen, watching the videos full screen and with constant volume.

The test persons were asked to first rate the aesthetic quality of the visualisation no and then the degree of accordance between music and visualisation. The first question was just used to assure, that the test persons do really just judge the degree of accordance in the second question.

3.3 Sample of test persons

Our sample of test persons consisted of 19 people in the age between 23 and 58 years. About half of them was female. About two third of the test persons had a high affinity to music, about one fifth a high affinity to graphics, both self rated a partly overlapping.

Given that sample, it is obvious, that the rating of the music as well as the basic elements for the visualisation is not universally valid in any way. Nonetheless, the ratings were significant for our mapping, because the same people were interviewed in the second round, so the mapping was built on top of their personal emotional connotations.

3.4 Bias

Every translational motion in the animations went from the bottom left to the upper right corner. Each rotation went clockwise.

The animations, song parts and visualizations with music were presented to each test person in a random order.

Because we used only midi files, the lyrics of the original song should not have a relevant influence on the ratings. Furthermore we only took western pop songs in the broader sense to avoid further complications because of cultural differences in music. However, the usage of quite popular songs could have had an unwanted influence on the rating.

Two of our test persons could only take part in the first round of the user study. There are no answers of them available from the second round.

3.5 Results of the user study

The next sections contain a more or less detailed evaluation of the ratings of the animations in the user study separately for each single visual parameter mentioned above.

Ratings of animations

Shape

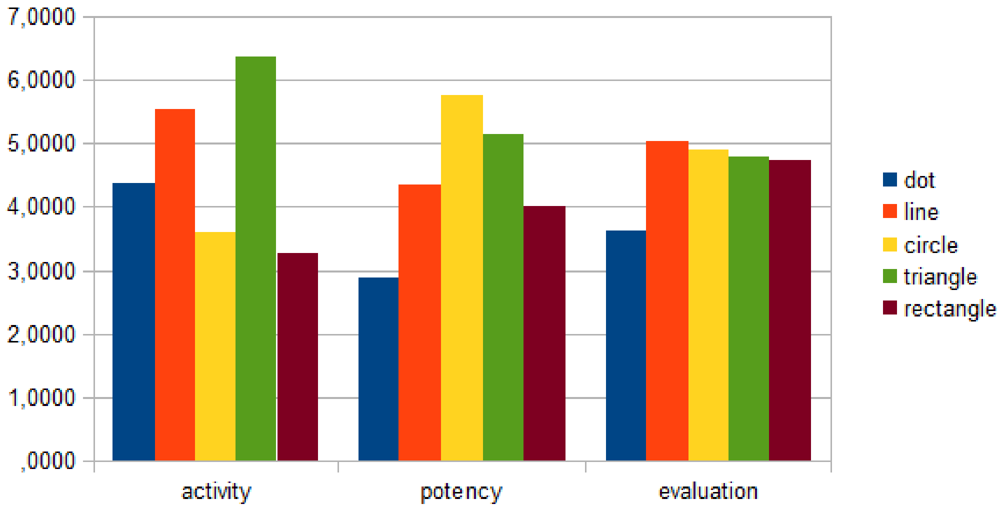

Figure 3.1: User study results based on shape.

Figure 3.1 shows, that dots are rated significantly worse than objects of other shape on the axes of potency and evaluation. Circles seem to be perceived more potent, but less active than other shapes. Triangles are rated far more active than any other shape.

Color

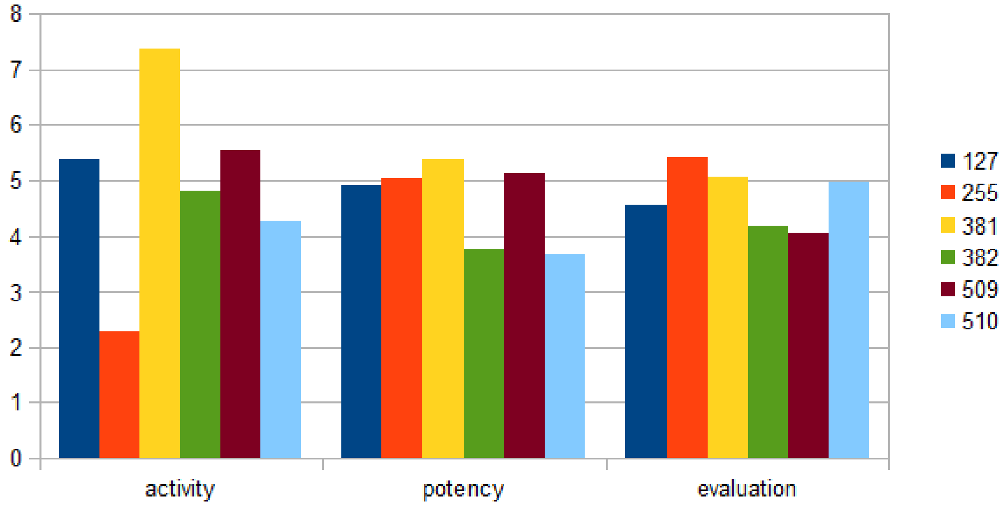

Figure 3.2: User study results based on brightness.

The most obvious conclusion with regard to the overall brightness of the objects used is, that brighter objects seem to be rated less potent than darker ones. Brightness was defined as the sum of the red, green and blue values as a value between 0 (black) and 765 (white).

This parameter is quite hard to interpret. There is no obvious correlation between the proportion of red and the ratings given by our test persons.

Objects with a higher proportion of green are obviously rated more potent and also more active than objects with a lower proportion of green, whereas there is probably no significant influence onto the evaluation axis.

Objects with a higher proportion of blue are obviously rated less potent than other ones.

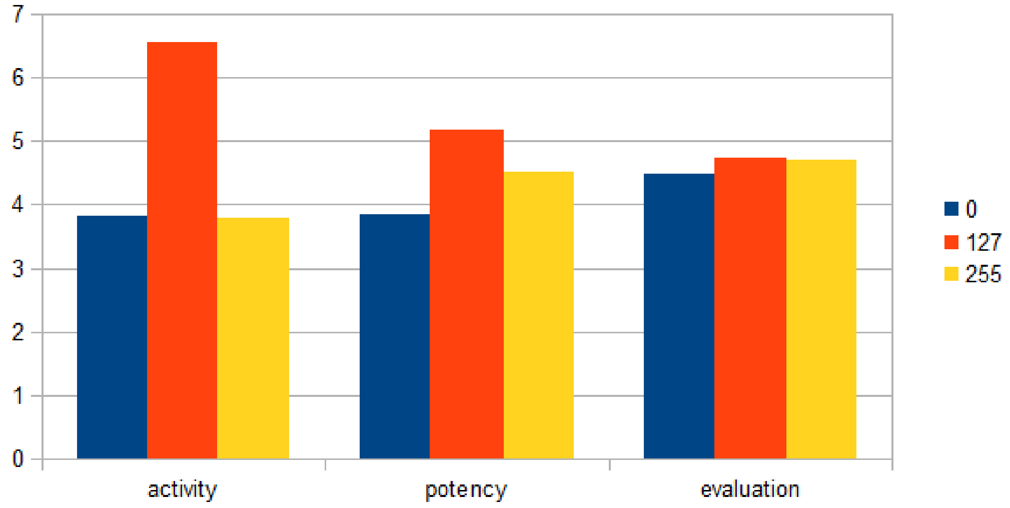

Figure 3.3: User study results based on portion of red.

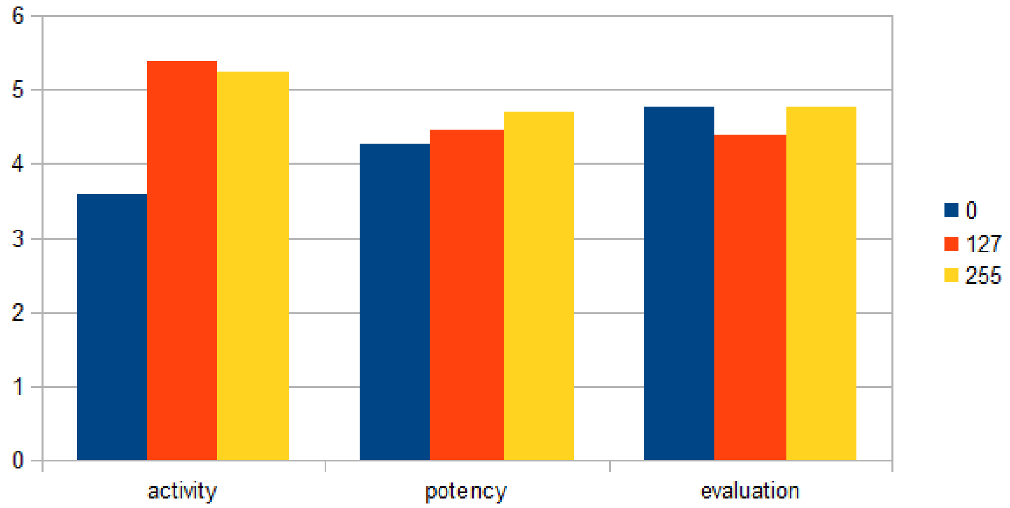

Figure 3.4: User study results based on portion of green.

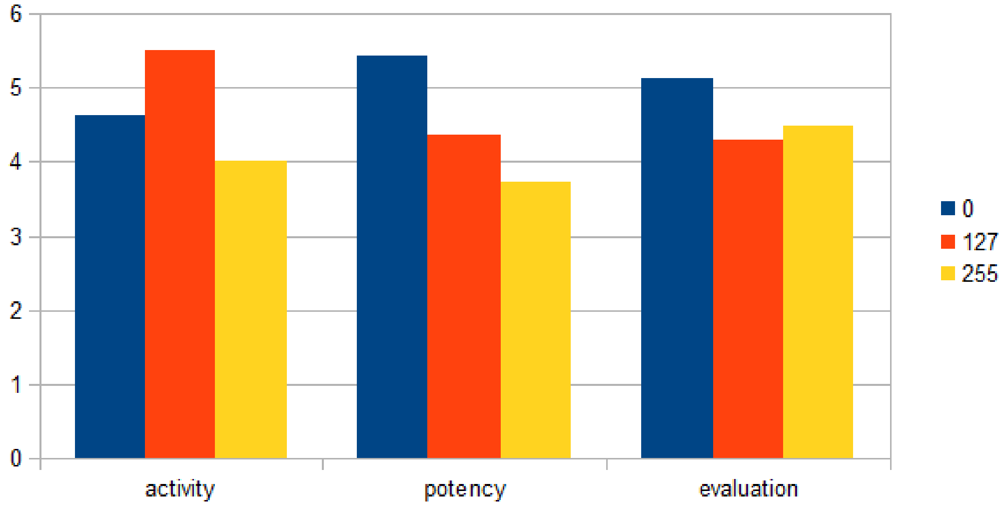

Figure 3.5: User study results based on portion of blue.

Number of objects

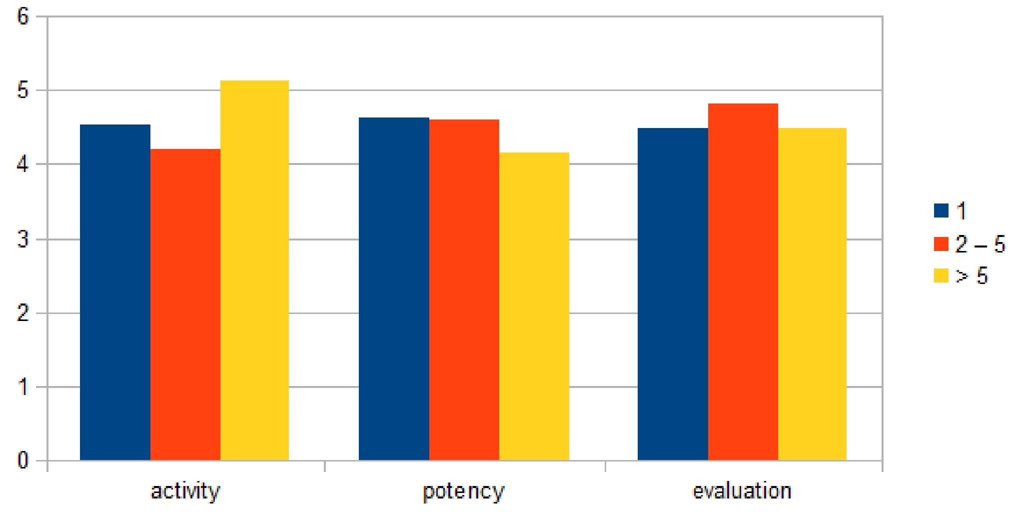

Figure 3.6: User study results based on the number of objects.

There is no obvious correlation here, either. However, the ratings on the potency axis surprisingly get slightly worse when more objects are used.

Size

Two quite obvious correlations can be found taking a deeper look onto the ratings based on the size of objects used. The potency strongly correlates to this parameter. The evaluation does, too.

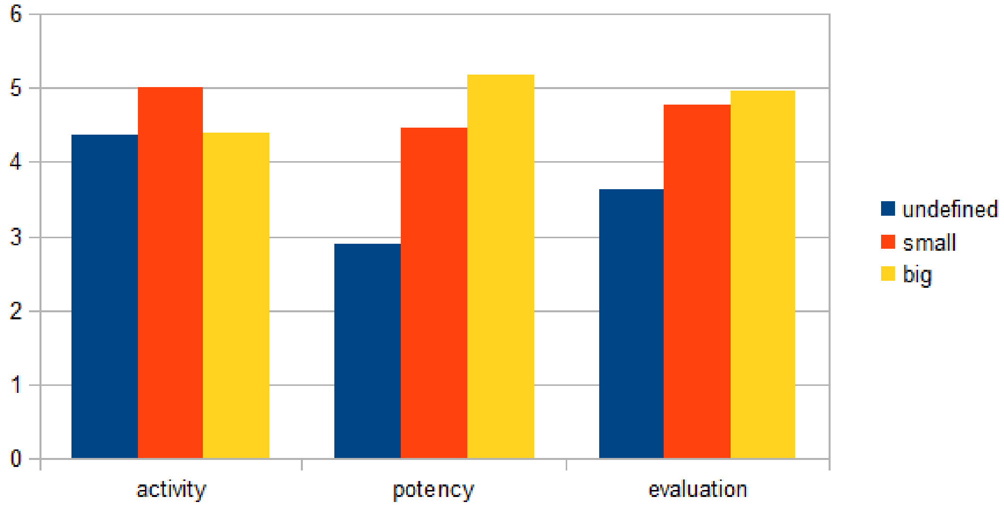

Figure 3.7: User study results based on size.

Motion

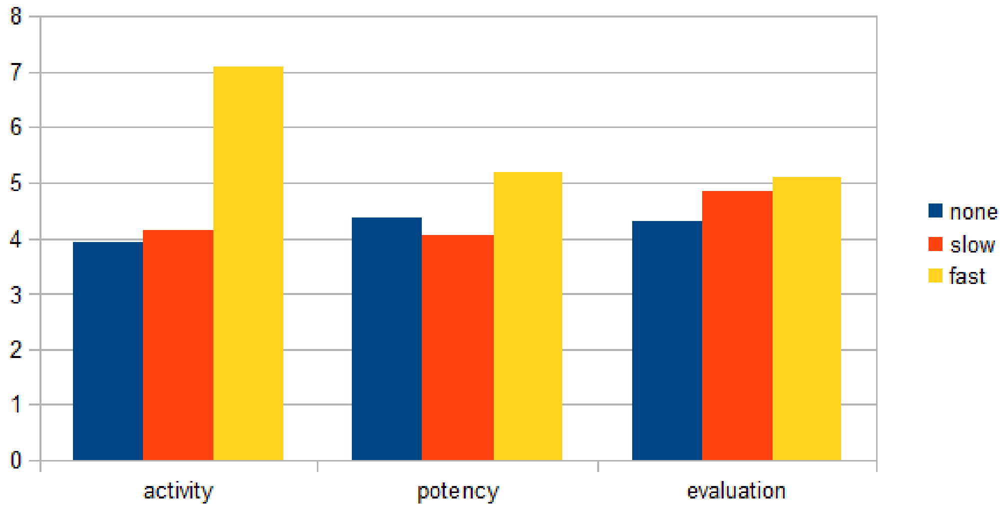

Figure 3.8: User study results based on the speed of rotation.

Fast rotating objects are obviously perceived much more active. Interestingly the difference in the average rating of slowly rotating objects and such ones not rotating at all is significantly minor. There is also a quite constant rise in the average rating when looking at the evaluation axis. The faster the objects rotated, the higher they got rated.

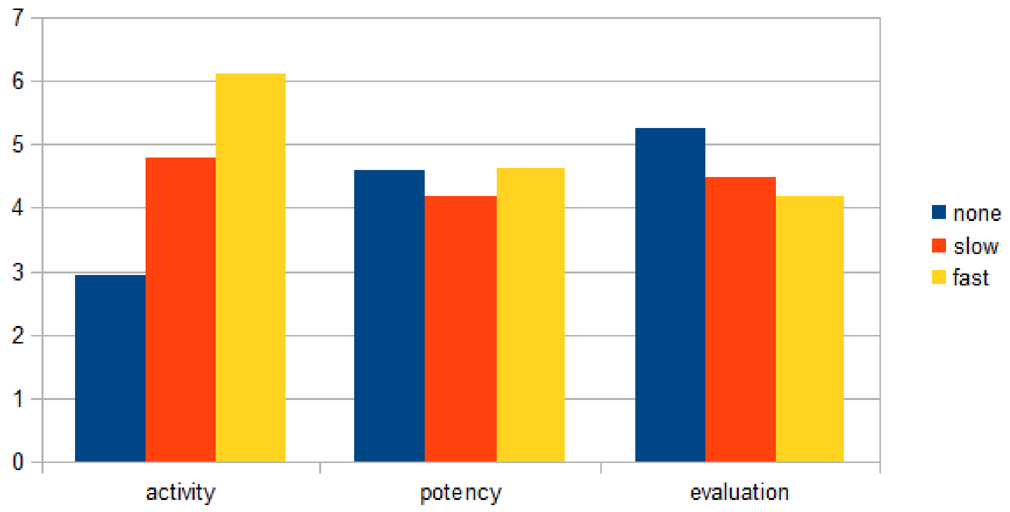

Figure 3.9: User study results based on the speed of translation.

It is interesting, that rotation as well as translation are rated higher on the potency axis when the speed is either null or fast. Slowly moving objects are rated lower in both cases. Furthermore, the translational speed is obviously strongly coupled to the rating on the activity axis, the faster the objects moved, the more active they were perceived by the test persons. More surprisingly, there is also a strong correlation between the evaluation axis and the translational speed. Whereas objects rotating faster were rated higher, there is a negative correlation between the translational speed and the average rating on the evaluation axis. In other words, the faster the objects moved, the worse they were perceived.

Conclusion

Activity Evaluating the ratings given by our test persons of the user study, we found out that, they rated animation more active, when fulfilling one or more of the following premises:

- triangles are being used
- no circles or rectangles
- the objects have a high proportion of green
- the objects are moving fast, either in a translational or a rotational way

Potency Animations were rated more potent dependent on the following premises:

- no dots are being used
- the used objects are darker
- used objects are less blue, more green
- there are not many objects used
- the objects used are bigger
- objects either do not move at all or at a high speed

Evaluation On the evaluation axis, animations got rated higher dependent on the following premises:

- no dots are being used
- the objects used are bigger
- objects are rotating on a higher speed
- the is only translational motion on a slower speed or none at all

Ratings of songs

Tempo

Looking at the results of the song rating based on tempo there is no doubtless correlation between the tempo and the three axis used. However, faster songs seem to be perceived tendentially more active and escpecially more potent than slower ones.

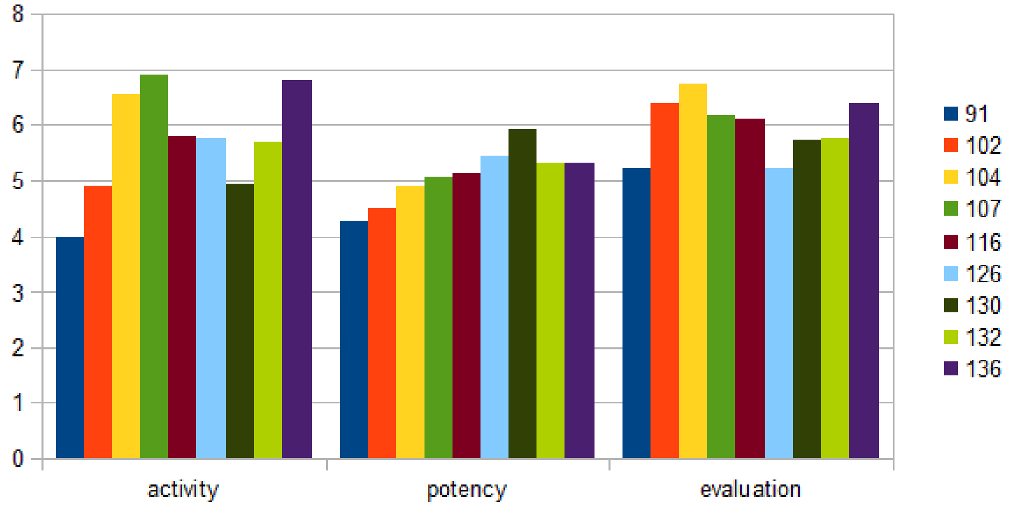

Figure 3.10: User study results based on tempo.

Instrumentation

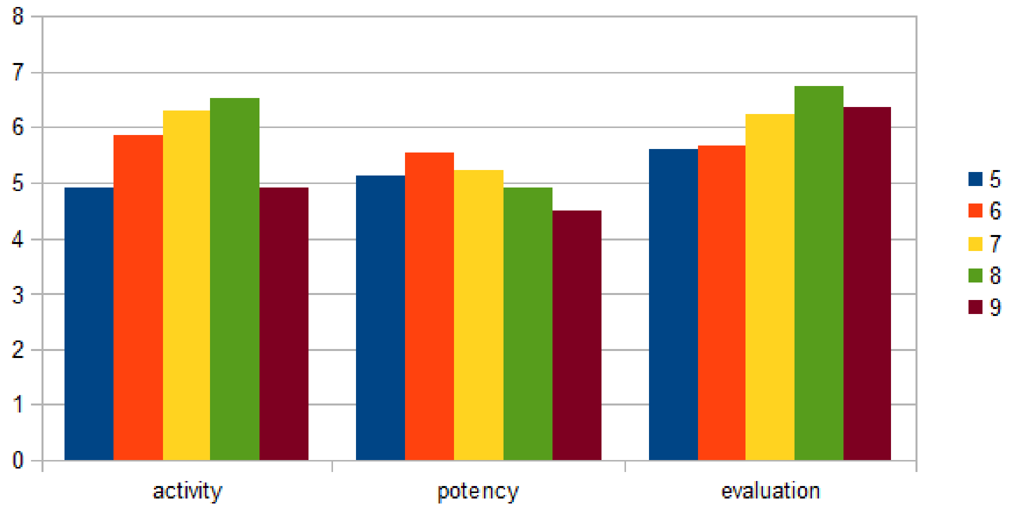

Figure 3.11: User study results based on the number of instruments.

More interesting than the evaluation of the tempo based ratings is the evaluation of the ratings based on the number of used elements. Obviously songs using more instruments tend to be perceived less potent and more positive. Cutting out the rating of the highest level (nine instruments) there is also a strong correlation between the number of instruments and the activity rating. More instruments seem to make a song more active based on the ratings of our test persons.

Mode

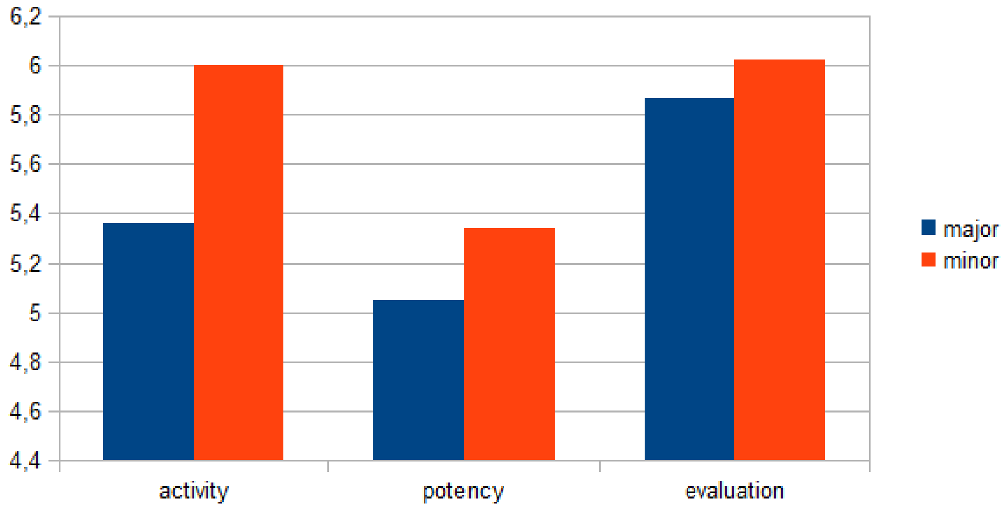

Figure 3.12: User study results based on mode.

Songs in minor mode are obviously rated higher in terms of activity, potency and evaluation than songs in major mode.

Conclusion

Activity Based on the ratings of our test persons, songs seem to be perceived more active when one or more of the following premises are fulfilled:

- eventually faster tempo
- eventually more instruments
- minor mode

Potency Songs were rated more potent dependent on the following premises:

- faster tempo

- fewer instruments
- minor mode

Evaluation On the evaluation axis, higher ratings have been given dependent on the following premises:

- more instruments
- minor mode

3.6 The mapping

Syntax

As we already described above, the mapping component of the application takes the track description created by the analyzer and the mapping file created by the user to merge both into the program to be given to the visualizer.

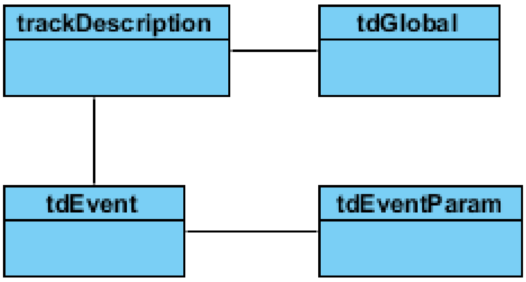

Figure 3.13: Allowed elements in the track description.

The basic syntax of the track description, illustrated in figure 3.13, is described as XML Schema. Here is a list of the elements used in this file:

- trackDescription
- tdGlobal
- tdEvent
- tdEventParam

The <trackDescription> element is the root element of the whole track description. It contains various <tdGlobal>-Elements. Each of those has to have a “name” and a “globalValue” parameter. Using this typical key/value-approach we are able to provide information about the track as a whole. Furthermore, the <trackDescription> element contains a number of <tdEvent> child elements. Each of those must have exactly two attributes: “time” and “name”. The “time” attribute explains, which kind of event happened, the other one - the “time” - indicates when it happened in milliseconds from the beginning of the track. Inside every <tdEvent> element there are one or more <tdEventParam> elements with the two attributes “paramName” and “paramValue”. Here we have a classical key/value approach, again. These event parameters give us detailed information about the event.

The global parameters currently provided by our prototype are:

- tempo
- timeSignature
- tonality
- mode
- trackLength
- instruments

The tempo parameter should be self-explanatory. As usual it is measured in beats per minute.

Time Signature describes the beat used, e.g. 3/4 or 4/4.

Tonality and mode, both give us information about the mode used in the song. The information provided by tonality is more detailed, whereas mode just says, if the tonality used is major or minor.

The trackLength pretty surprisingly provides information about the length of the track, in milliseconds again.

The last parameter, instruments, contains the number of instruments used in the song/track.

The basic syntax of the mapping file is also described as XML Schema. The basic elements to be used in the mapping are:

- mapping
- pattern
- event
- command
- param

The following figure illustrates the relationship between these element types:

As figure 3.14 clearly shows, the <mapping> element is the root element of the whole mapping structure. It does not provide any attributes.

Inside the <mapping> element, one or more <pattern> elements can be used, each of them with the obligatory attribute “name”. Exactly one of them has to be named “global”. The current version of our prototype only uses this one. However, we wanted to have a structure for flexible

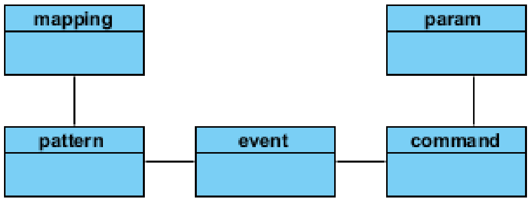

Figure 3.14: Allowed elements in the mapping file.

in the mapping schema to allow future versions of the application to also support patterns like explained in the chapter "future work".

Within each <pattern> element, one or more <event> elements can be used. Those Events are identified by the "name" attribute. The events currently supported by the prototype of our application are:

- pause
- note
- breakStart
- breakEnd
- chord

The names of the events provided are rather self-explanatory. Each of these events supports an individual set of input parameters. The parameters provided by the "note" event are:

- key
- volume
- duration
- frequency
- pan
- pitch
- instrument
- channel

The "pause" event has the following parameters:

- duration

- instrument
- channel

The parameters of the “chord” event are listet here:

- chordName
- chordType
- instrument
- channel
- duration

The “breakStart” and “breakEnd” events do not have any parameters.

These parameters can be used with the next two elements, the <command> element, which is situated directly under the <event> element, and the <param> element, which is situated one level below and can be seen as the element used as configuration for its parent <command> element.

The <command> needs to have at least the name of the command class, that should be called whenever the mother <event> fires. However, the name of the command may be prefixed with an integer n followed by a “:“. That means, that this command will not only be executed once, but n times when it is triggered by the event. The integer value does not have to be hard coded, but can also be the result of a calculation explored in more detail later on in the section “calculating attribute values”. The value of the “name” attribute has to match exactly the name of a class within the visualization package.

The parameters needed differ from command to command. Each parameter is represented by the corresponding element within the mother <command> element. It has to have a “name” attribute - the key of the expected parameter - and a “paramValue” attribute. The latter also supports the calculation syntax mentioned before and explained in the next section.

Calculating attribute values

To allow more dynamic mappings, it is possible to not only use numerical values with the prefix of command names or within paramValues, but to calculate these values using dynamic data from different sources.

The least dynamic way to do this is to use theme data provided by the theme configuration file. The data provided differs from theme to theme, the syntax is quite easy, e.g.: paramValue=“+theme.anyvar”

It is also possible to access global track information in a similar way. To get the track length, you just have to use “+track.trackLength”. The tempo of the current track or song can be accessed with “+track.tempo”.

The most dynamic data available is provided by the current event. For example, you can use “+event.key” or “+event.volume” to use detailed information of a note event to calculate the input parameters for a command triggered by that event.

In many situations, the value of a command parameter will not only be dependent on one single input parameter, but on a couple of them. Therefore it is possible to connect them using the “|” symbol. To give an example, let us assume, that a command parameter has a value of “+event.frequency|+123|+track.tempo”. In that case the value given to the command for that parameter will be the sum of the frequency of the current note, the number 123 and the tempo of the song.

Furthermore, the user is not limited to only use addition to calculate parameter values, but is also allowed to use the symbols “*” for multiplication and “-” for subtraction. However, the calculation is always done from the left to the right, not caring about the rules of priority in mathematics.

At least there are some reserved words with special functions. “random” provides a random number between 0 and 1, including 0, but not including 1. “lastId” provides the id of the last primitive created. That way, it is possible to apply some kind of operation immediately after creating it, for example to create a primitive, which should start to move immediately after its creation. Every value starting with Cb is interpreted as a callback name, which causes the application to look for a Callback class with exactly that name and using the return value of the class as the attribute value. Callbacks can contain any java code and are therefore ideal for implementing complex mapping rules or mapping based on non-numerical decisions, e.g. you are dependent on enumerations with values in no particular order.

CHAPTER 4

Implementation

4.1 Outline

When there are relationships between color and music, and emotion is the mediating element between them, there are probably also systematic relationships between music and other visual elements. Those relationships shall be used as a basis for a technical solution providing a better user experience in automated music visualization.

Therefore we tried to figure out these relationships using the results of our user study explained in the chapter "Methodolical Approach". Using visual elements with the same emotional connotations as the music they are coupled to, should result in a visualization feeling more natural than common solutions.

To test our thesis, we implemented a kind of prototype for a visualization software working on this basis. This prototype is meant to be a low level implementation of our thesis and can be refined and enhanced in a couple of ways as we will show in this chapter.

4.2 Functionality

As shown in figure 4.1, our prototype consists of three bigger parts. A detailed technical documentation of our prototype can be found in the appendix.

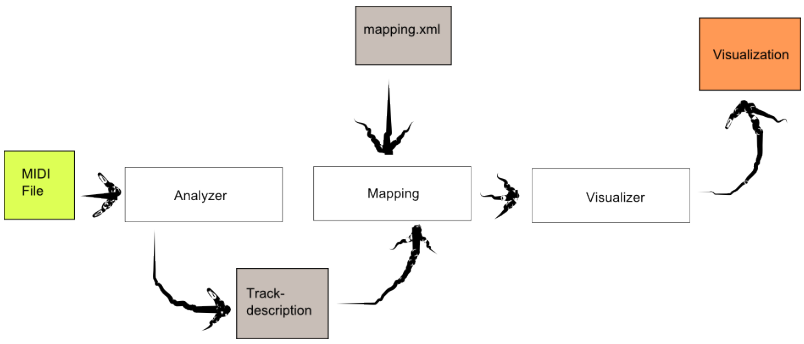

Figure 4.1: The architecture of our prototype.

Analyzer

The first of these three components is the music analyzer. It is used to automatically read out as many information as possible from a given piece of music. A final version of the application should contain at least two different implementations for the analyzer component, one dealing with MIDI files, the other one dealing with WAVE files. In our prototype version we provide an abstraction layer allowing to extend the basic functionality by implementing own analyzer classes. We only provide a single analyzer implementation, the one dealing with MIDI files. The decision for MIDI and against WAVE was made purely pragmatical. It is much easier to get enough of the information we need for our visualization out of MIDI files than it is to get it out of WAVE files.

The analyzer component reads out as much information as possible and puts it into a standardized XML form. A more detailed documentation of this component can be found in the bachelor thesis of Alexander Hauer. [5]

Mapping

We call the information provided by the analyzer as XML file the track description. This track description is one of the sources needed by the second component of our application, the mapping component. The second source needed by the mapping component is the mapping file, which is also an XML file. Whereas the track description contains a number of global information parameters and numerous events with time stamp and more detailed parameters, the mapping file contains concrete instructions, which acoustic event triggers which visual result and how global parameters of both worlds are coupled. The aim of the mapping component is to provide detailed orders to the visualizer, the third part of our application. Therefore it has to generate a program for the given music file, that contains detailed information when which command provided by the visualizer has to be triggered and how the input parameters taken form the track description should be mapped into arguments for the command. More detailed information about the format of the two files, the track description and the mapping file, is provided in the next chapters.

Visualizer

The visualizer is the third part of our application and responsible for the final visual output. It gets its orders from the mapping component and in return provides the visual elements the mapping refers to in its orders. Similar to the analyzer component, the visualizer also provides a kind of abstraction layer. In our prototype version, we provide a quite minimalistic and generic theme allowing the use of graphical primitives like lines, circles and triangles.

The application architecture allows the use of any theme implementing a rather simple interface. More detailed information about that can be found in the bachelor thesis of Andreas Schmid. [11]

4.3 Relationships between acoustic and visual elements

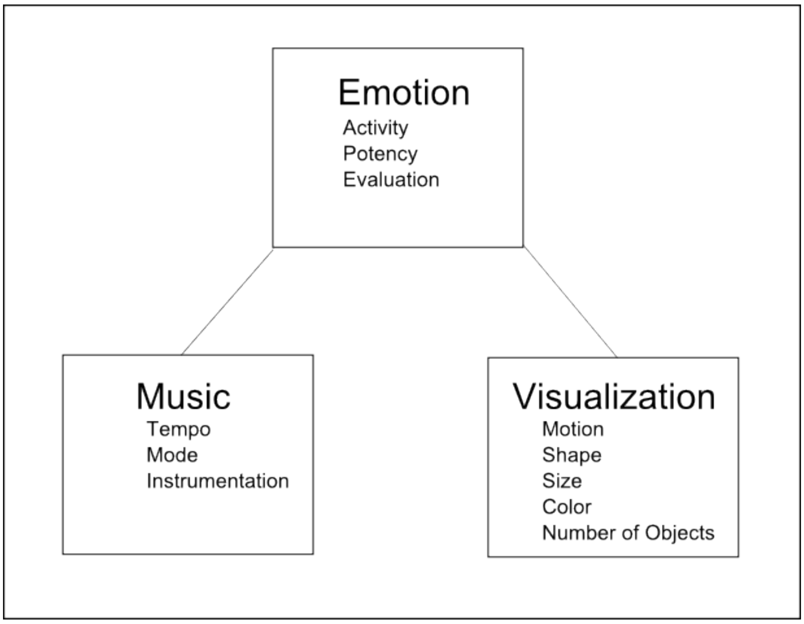

Figure 4.2: Interrelations between acoustic and visual elements.

Figure 4.2 shows the interrelation between the acoustic and the visual elements our mapping relies on. Each of these elements can have weaker or stronger emotional connotations. As Palmer explains, those connotations are often caused by a semantic meaning based on personal or collective experience, that is coupled with the element. [12] As explained in the previous chapter, these elements used as input and output parameters build the foundation of our work. To keep it simple, we decided not to support the visualization of WAVE files of any kind (e.g. WAV, MP3, OGG and others). On the other side we also had to limit our effort, so we broke down the exemplary visualization theme of or prototype to the usage of basic primitives. However, the methodological approach can be reused working with much more complex themes and of course there is no general limitation on the analytical side, either. Later on, we want to take a deeper look onto the parameters not used in this work. But first we want to explain in detail, which parameters have been used and which units or categorisations were used for each of them. Figure 4.3 gives a quick overview.

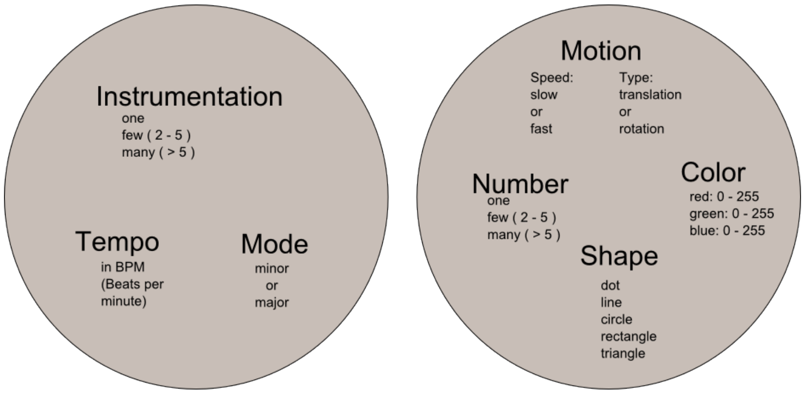

Figure 4.3: Input and output parameters of the prototype.

MIDI parameters

The parameters provided by our MIDI analyzer are rather simple. We used the tempo and the instrumentation, which both can be more or less read out directly from the MIDI data given. The use of BPM (beats per minute) as unit for the tempo is evident. It was also quite practical to use the raw number of instruments used in the song. When working with WAVE files, it would probably also make sense, to differentiate between the different classes of instruments because of their very typical sound. For the sake of simplicity, we decided to ignore that in our case. To get the third parameter out of a MIDI file, a bit more work was necessary. As explained in the bachelor thesis of Hauer mentioned above, we needed to write an algorithm to determine the most probable key and tonality, due to the fact, that it is not clear in every single song or track.

Furthermore we read out a couple of global parameters like e.g. the track length. We also defined some parameters, that can not be provided by a MIDI analyzer, but should be read out when implementing an alternative WAVE analyzer.

Based on the information we got from the analyzer, we were able to create mappings using psychological knowledge, that take care of the emotional connotation of music and visual elements. Those mappings should be able to produce visuals feeling more natural on the fly given any valid MIDI file. Of course, the cultural background of the user must not be ignored. Therefore our exemplary mapping does not claim general validity, but is specifically targeted to an audience represented by our sample of test persons.

Because of the layering of our application, it is pretty easy to implement specific mappings for different user groups and even implement own themes with focus on specific genres or topics. However, the mapping always depends on the theme used. More information to the development of own themes is given in the appendix. The syntax and the development of mapping files is explained in the next sections of this chapter.

Visual parameters

On the other - the visual - side, we used some more parameters for our mapping. Besides the color parameter, which will be discussed in more detail in the following section, we used motion, size and number of elements as well as their shape. The shapes we used were the basic elements dot, line, triangle, rectangle and circle. With regard to the motion, we took into account the type of motion (transitional or rotational) and the speed (slow or fast). Further more we cared about the number of elements (one, two to five, more than five) and their size (big or small). It seems natural, that some combinations just do not make any sense like rotating circles or dots or trying to measure the size of a dot.

Colors

A good example for the cultural differences in the meaning of parameters is the usage of colors in different cultures. As mentioned in the chapter "State of the art" there are common associations with colors across all over the world. However, there are specific meanings connoted with certain colors in certain cultures. Also there are sets of colors often used in specific cultures (e.g. orange/brown in funk or soul), but feeling quite foreign in other cultures (e.g. heavy metal).

Another specific aspect concerning colors is the way they are classified in the digital space. Each color has its specific position in the RGB (red, green, blue) color space. Having the discrete values of each of these three basic colors, it can further more be classified based not only on its hue, but also on brightness and saturation. Therefore we decided to split up the color component into these three values to be able to do mathematical operations, both during the evaluation of the user study as well as on run time when mapping acoustic parameters onto colors.

4.4 Prototype

This section will provider a technical explanation of the mapping component of our prototype. More detailed information about the other two parts of the application are given by Alexander Hauer and Andreas Schmid ([5], [11]). The whole source code of the prototype implementation can be accessed cloning the GIT repository publicly available at https://bitbucket.org/juergen.giefing/physics-based-music-visualization.

Figure 4.4 illustrates in a simplified way the most important classes of the mapping component and the interrelations between them.

Basically the program for the whole visualization is computed instantly after the analysis of the music file by the analyzer. The visualizer class is getting created, takes the mapping file and the track description and calls its own initVis() method. In this method the Visualizer iterates through the track description. At every step it looks up the mapping file for a concrete directive matching the current event. If one or more of these directive exist, it chronologically reads out every command associated with the event and calculates the command parameters.

After creating the appropriate command using reflection, the command including the time stamp of the event is saved into an internal list.

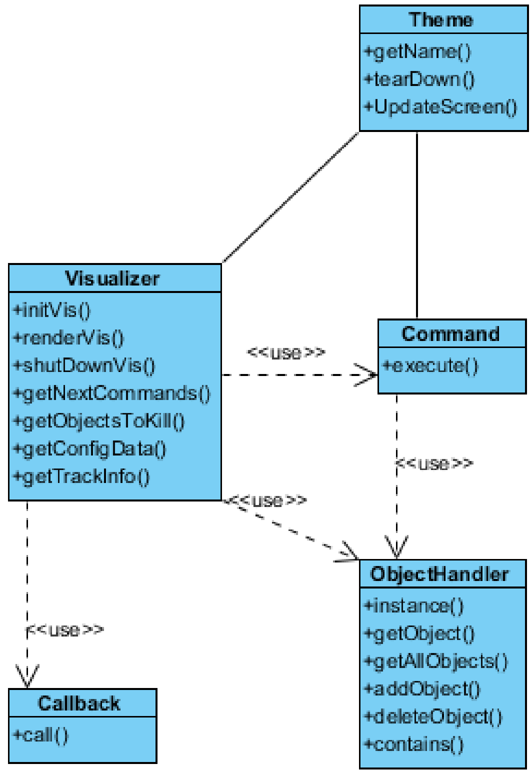

Figure 4.4: The most important classes of the mapping component and their interrelations.

As soon as the whole mapping is created, the visualization process itself starts. In a loop, the whole program list is iterated through. Each iteration starts getting all commands, that have to be executed until the next frame. These commands are determined by getting all commands with a time stamp between the current time plus the milliseconds used for a single frame. This number is controlled by a configuration option, that indicates the frames per second. The commands are executed one after another as they are stored in the internal list.

Objects created during this process are automatically stored in the object handler, where they are available all the time for every command and the visualizer itself. That way it is quite easy to create an object using a creation command and immediately apply another command.

The second action beside the execution of the commands performed in each iteration is the deletion of all objects, that have reached their time to live calculated at the time of their creation.

The theme has to provide all commands used in the mapping. It also can provide some own configuration and is responsible for the rendering itself by implementing the updateScreen() method and the tearDown() method called at the end of the visualization from the shutDownVis() method of the visualizer.

Command is another abstract class used by the visualizer when it calculates the commands parameter values. More precisely the visualizer uses the concrete command implementations, which are referenced by name in the mapping and are, like commands, also created dynamically by reflection.

CHAPTER 5

Results

5.1 User Study, Part 1

Exemplary Mapping

Based on the results of our first user study, we built up the following table, giving an overview of our conclusions:

-	shape	translation	rotation	brightness	red	green	blue	size	number
fast	tri	fast	fast	dark		more		big	
slow									
more instr.	tri / dots	fast / slow, no	fast			more	more	big	many
less instr.									
major	circ, rect, dots						more		many
minor	tri	fast / slow, no	fast	dark		more		big	

Table 5.1: Conclusions based on our first user study

Obviously we have a contradiction in the translation column, saying that songs in minor songs with more instruments should be mapped to animations using objects rotating slow, fast and not at all at the same time. So we decided to ignore this column. The second column we decided not to use was the “red column”, because there was no relation to any audio parameter at all. Every other column had at least one piece of information useful for our mapping.

Shape

The shape used to create a new primitive is decided in the CbShape callback class. There we decide based on the tempo, the instruments and the mode, whether we create a dot, a line, a circle, a rectangle or a triangle. There is also a bit of random in the algorithm.

Translation

Because of the contradictions mentioned above, we decided to use just random values for the translational speed of our primitives.

Rotation

To make the rotation dependent on the mode of the song, we used the directive "*0|+track.mode|*360" as value of the "angle" attribute. The speed is controlled by the number of instruments used and the tempo, using the directive "-track.tempo|*0.1|-track.instruments|*-0.07|*-1|+2".

Brightness

Brightness is dependent on the tempo and the mode of the song, the faster the song is, the darker the colors should be. Minor songs are perceived darker, too. Because darker color means less of each of the three basic colors in RGB color space, we first thought about using lower red, green and blue values when calculating the colors of our objects. Finally we chose another approach to keep the mapping more readable. Because of the black background, more transparent objects unavoidably appear darker. Therefore we used the alpha channel to control the brightness of our objects. The directive used is "*0|+track.mode|-track.tempo|*0.01|+2";

Red

Analogically to the values used for translational motion, we decided to also use the random function here.

Green

The portion of green is dependent on the number of instruments (the more instruments, the more green), the mode (more green in songs in minor mode) and the tempo (the faster, the more green). Therefore we decided to use that directive:

"-0.8|*random|+0.9|+2|-track.mode|*40|+track.tempo|*0.3|+track.instruments|-30|*0.02".

Blue

The amount of blue is dependent on the mode (major lead to more blue) and the number of instruments (the more instruments the more blue). This is the directive we used:

"-0.8|*random|+0.9|+track.mode|*0.05|-0.5|+track.instruments|*0.15".

Size

The size of objects should depend on the number of instruments used (the more instruments the bigger), the mode (bigger objects for songs in minor mode) and the tempo(the faster the bigger). Therefore the directive we used for both parameters, sizeX an sizeY, was

“+track.tempo|*0.15|+track.instruments|*0.15|-track.mode|*0.03”.

Number of Objects

The number of objects is dependent on the mode of the song and the number of instruments. Therefore we used the following directive: “+track.mode|*track.instruments|*0.05|+2”.

Anti-Mapping

The Anti-Mapping was created using random values for every parameter given to achieve a totally random behaviour.

5.2 User Study, Part 2

The evaluation of the second part of our user study clearly shows, that at least this simple approach was not successful. The visualizations of all common players got better rated in terms of aesthetics as well as accordance between music and visualization, as can be seen in table 5.2 and figure 5.1.

As mentioned in the chapter about our methodological approach, the main purpose of the first question, concerning the aesthetic appeal of the visualization was to take this aesthetic factor out of the answers given to the second question as far as possible. The second purpose was to see possible correlations between the two axis.

However, there seems to be a weak correlation between them. In both ratings, iTunes got rated best and the visualizations based on our mapping got rated worst. Nevertheless, this does not mean, that the test persons misunderstood the two questions. The proportion between the rating in terms of the first question and the rating in terms of the second question strongly differs between the single players, which shows, that the aesthetic perception does obviously not strongly influence the answers given by the test persons when they were asked about the accordance between the visual and the acoustic part of the video.

	Aesthetics	Accordance
WinAmp	3,4176	2,9941
WMP	3,0471	3,0882
iTunes	3,8353	3,4000
Mapping	2,5000	2,7882
Anti-Mapping	2,8353	2,9765

Table 5.2: The results of the second part of the user study on a scale from 1 (worst) to 5 (best)

Particularly interesting is the fact, that even the visualizations based on the anti-mapping were rated better than the visualizations based on the mapping.

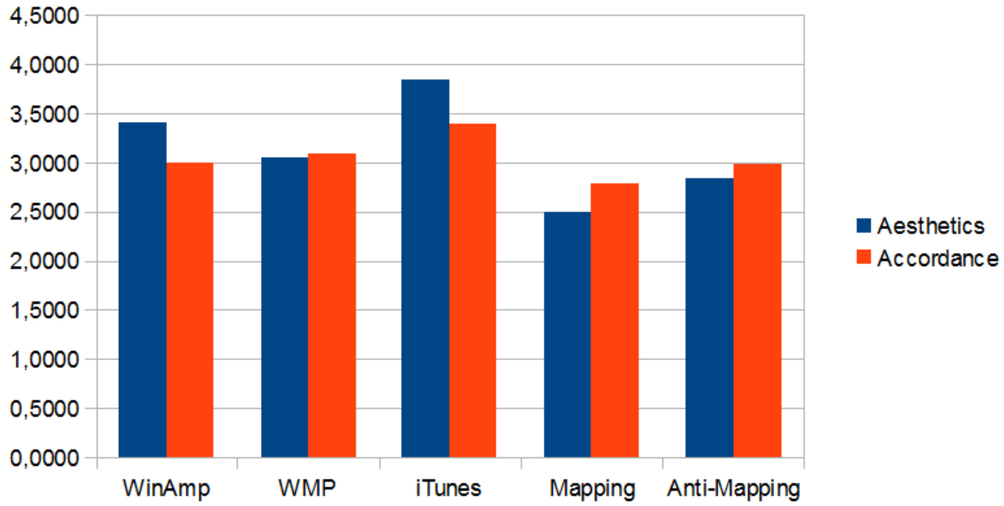

Figure 5.1: The results of the second part of the user study as a diagram on a scale from 1 (worst) to 5 (best)

CHAPTER 6

Summary And Future Work

6.1 Conclusion

There might be various reasons for that result. The correlation between te aesthetic perception and the accordance mentioned in the Chapter “Results” might be a minor part cause, but not the essential one.

The most obvious explanation for the result might be, that our approach was simply wrong. Regarding the study and the context in detail shows, that it maybe could be refined in a way, that it works.

We would suggest to keep the principle of mapping sound to emotion and emotion to visuals. But it seems to be crucial to refine the mapping process.

Our approach was a pretty simple one. We just mapped single attributes to parameter values not regarding the combination of attributes and the dependencies between then. It might be crucial to replace theses low level mappings by more complex ones, based on a number of - maybe fuzzy - definitions. This way would allow mappings like “use bigger and greener primitives if the song is pretty fast and in major mode, but only if there are more than five instruments”.

Even more promising is an approach introducing an additional layer of abstraction to directly map audio parameters and their relationshipts to each other to predefined sets of mood and at the same time map those to interdependent parameters for the visualization. Therefore the probably most important improvement would be the implementation of pattern recognition as explained below. That way, the rudimentary analysis of song structures and the low level approach to combine the musical information with emotion could be put onto te next level.

Another possibility worth trying to get a more satisfying result could be the development of a more sophisticated theme. If there are enough capacities to provide a theme with more complex and more powerful commands could help to build a better mapping. Our exemplary theme only provides very limited possibilities to visualize the song resulting in rather similar visualizations for all our sample songs.

Additionally the user study process itself could be improved regarding the issues mentioned in the chapter “Bias”.

6.2 Future Work

Theme Development

Because of the layered architecture of our application it’s rather simple to develop completely new themes providing own commands, the user can use in his mapping. The theme is only loosely coupled to the visualizer, so there are almost no limits regarding the primitives used by the theme or even the underlying concept of the visualization approach.

Wave Analyzer

A pretty obvious enhancement of our prototype application would of course be an analyzer dealing with any kind of wave file, most useful would probably be a mp3 analyzer. Such an analyzer would allow to consider audio information not available in MIDI files because of obvious reasons. Especially the timbre and the volume variations might have enormous influence on the perception of music.

Pattern Recognition

The current prototype provides rudimentary support for the evaluation of patterns in songs. Patterns are also part of the syntax of the mapping file. However, the current midi analyzer does not really recognize special patterns in input files, nor does it support recurring structures. Both, patterns - defined either using hard boundaries for parameter values or in a more or less fuzzy way - and recurring structures, could help to put the mapping approach onto a higher level.

Patterns could be used to bring more semantics into the mapping. A concrete mood could be defined combining a number of constraints containing every available data extracted from the audio file. Based on psychological knowledge, this could lead to even more natural visualizations, because parameters would not be considered separately any more, but also with regard to their interrelationships.

Combined with the recognition and consideration of song structures like verse, chorus, bridge and others, the structure of a song could be considered as a whole and recurring acoustic parts have their equivalents in the visualization.

Further audio parameters and events

The midi analyzer does already extract the most important information and a number of potential types of events from the given file. However, there is still many information not used yet. At the moment, we only differentiate between the mode - major or minor - although it would already be possible to go one step further and regard the specific tonality. The main reason, why we do not do this yet, is the fact, that it is quite hard to use this information without having further information. Basically it does not matter, if a song is in c major or g major. What does matter is the harmonic progression, how melody behaves regarding the chords behind, how chords are varied and similar aspects in terms of harmony. The whole theory of harmony could be used to get much more information out of the given data. The basis for that - the analysis of chords - is already implemented in our current prototype.

Colors

In our exemplary mapping we only regarded the single proportions of red, green and blue and the brightness. There are a lot more possible approaches how to calculate the colors of the primitives regarding hue and saturation. These parameters might also have given additional insights when used in the first part of the user study.

CHAPTER 7

Appendix

7.1 User Studies

Sample Of Test Persons

The following table shows the composition of our sample of test persons with regards to their affinity to music. The scale was from 1 (no affinity at all) to 5 (maximal affinity). The next table shows the composition of the sample of test persons using the same scale but with regards to their visual affinity.

affinity	count
1	0
2	1
3	4
4	8
5	6

Table 7.1: Self assessment of the test persons concerning their affinity to music

affinity	count
1	0
2	4
3	8
4	4
5	3

Table 7.2: Self assessment of the test persons concerning their affinity to visual elements

sex	count
female	9
male	10

Table 7.3: Composition of the sample of test persons based on sex

age	count
20-29	5
30-39	11
40-49	1
50-59	2

Table 7.4: Composition of the sample of test persons based on age

Sample Of Song Parts

Title	Artist	Mode	Tempo	Instruments
Cryin	Aerosmith	major	102	9
Crying at the discotheque	Alcazar	minor	136	7
Eleanor Rigby	The Beatles	minor	130	5
Everyday People	Arrested Development	major	91	5
Pick up the pieces	Average White Band	major	107	6
Rain	Annie Lennox	minor	126	6
Respect	Aretha Franklin	major	116	7
School's out	Alice Cooper	major	132	5
Stayin Alive	Bee Gees	minor	104	8
TNT	AC DC	major	130	6

Table 7.5: Songs used in our user studies

Sample Of Animations

	red	green	blue	translation	rotation	shape	size	number
1	0	127	255	slow	none	dot	undefined	>5
2	255	0	127	fast	none	dot	undefined	2-5
3	255	0	255	none	slow	line	small	>5
4	127	255	0	slow	fast	line	big	2-5
5	127	127	255	fast	none	circle	small	>5
6	255	0	0	none	none	circle	big	2-5
7	0	127	0	slow	slow	triangle	big	1
8	127	127	127	fast	fast	triangle	small	>5
9	0	0	255	none	slow	rectangle	small	2-5
10	0	255	127	slow	none	rectangle	big	1

Table 7.6: Animations used in our user studies

Semantic Differential Scales

axis	minimum	maximum
activity	1 (not active)	10 (very active)
potency	1 (not potent)	10 (very potent)
evaluation	1 (very negative)	10 (very positive)

Table 7.7: Scales used in our user studies

Results

Ratings of animations

Shape

shape	axis	min	avg	max
dot	activity	1	4.3684	9
dot	potency	1	2.8947	7
dot	evaluation	1	3.6316	7
line	activity	1	5.5263	9
line	potency	1	4.3421	9
line	evaluation	1	5.0263	10
circle	activity	1	3.6053	10
circle	potency	1	5.7632	10
circle	evaluation	1	4.8947	10
triangle	activity	3	6.3684	9
triangle	potency	2	5.1316	9
triangle	evaluation	1	4.7895	9
rectangle	activity	1	3.2895	8
rectangle	potency	1	4.0000	9
rectangle	evaluation	1	4.7368	8

Table 7.8: Ratings based on the shape used in the animation

Color

brightness	axis	min	avg	max
127	activity	3	5,3684	9
127	potency	2	4,8947	8
127	evaluation	2	4,5263	8
255	activity	1	2,2895	8
255	potency	1	5,0263	10
255	evaluation	1	5,3947	10
381	activity	5	7,3684	9
381	potency	2	5,3684	9
381	evaluation	1	5,0526	9
382	activity	1	4,8026	9
382	potency	1	3,7895	9
382	evaluation	1	4,1974	9
509	activity	1	5,5263	10
509	potency	1	5,1053	9
509	evaluation	1	4,0526	7
510	activity	1	4,2632	7
510	potency	1	3,6842	8
510	evaluation	1	4,9474	10

Table 7.9: Ratings based on the brightness of the color used in the animation

blue	axis	min	avg	max
0	activity	1	4,6140	9
0	potency	2	5,4386	10
0	evaluation	1	5,1228	10
127	activity	2	5,4912	9
127	potency	1	4,3684	9
127	evaluation	1	4,2807	9
255	activity	1	4,0000	10
255	potency	1	3,7105	9
255	evaluation	1	4,4868	10

Table 7.10: Ratings based on the proportion of blue used in the animation

green	axis	min	avg	max
0	activity	1	3,5658	9
0	potency	1	4,2763	10
0	evaluation	1	4,7763	10
127	activity	1	5,3947	10
127	potency	1	4,4474	9
127	evaluation	1	4,3816	9
255	activity	2	5,2368	9
255	potency	1	4,6842	9
255	evaluation	1	4,7632	9

Table 7.11: Ratings based on the proportion of green used in the animation

red	axis	min	avg	max
0	activity	1	3,8158	9
0	potency	1	3,8289	9
0	evaluation	1	4,4737	8
127	activity	1	6,5614	10
127	potency	1	5,1579	9
127	evaluation	1	4,7368	9
255	activity	1	3,7895	9
255	potency	1	4,4912	10
255	evaluation	1	4,6842	10

Table 7.12: Ratings based on the proportion of red used in the animation

number	axis	min	avg	max
1	activity	2	4,5263	9
1	potency	1	4,6316	9
1	evaluation	1	4,4737	8
2 - 5	activity	1	4,1974	9
2 - 5	potency	1	4,6053	10
2 - 5	evaluation	1	4,8158	10
> 5	activity	1	5,1184	10
> 5	potency	1	4,1447	9
> 5	evaluation	1	4,4868	10

Table 7.13: Ratings based on the number of objects used in the animation

Number of objects

size	axis	min	avg	max
undefined	activity	1	4,3684	9
undefined	potency	1	2,8947	7
undefined	evaluation	1	3,6316	7
small	activity	1	5,0132	10
small	potency	1	4,4474	9
small	evaluation	1	4,7763	10
big	activity	1	4,3816	9
big	potency	1	5,1711	10
big	evaluation	1	4,9474	10

Table 7.14: Ratings based on the size of objects used in the animation

Size

Motion

speed	axis	min	avg	max
none	activity	1	3,9263	10
none	potency	1	4,3368	10
none	evaluation	1	4,2947	10
slow	activity	1	4,1754	9
slow	potency	1	4,0702	8
slow	evaluation	1	4,8421	10
fast	activity	4	7,0789	9
fast	potency	2	5,1842	9
fast	evaluation	1	5,0789	9

Table 7.15: Ratings based on the rotation of objects used in the animation

speed	axis	min	avg	max
none	activity	1	2,9474	8
none	potency	1	4,5789	10
none	evaluation	1	5,2456	10
slow	activity	1	4,7895	9
slow	potency	1	4,1711	9
slow	evaluation	1	4,4868	9
fast	activity	1	6,1053	10
fast	potency	1	4,6140	9
fast	evaluation	1	4,1579	9

Table 7.16: Ratings based on the translation of objects used in the animation

Ratings of songs

Tempo

tempo	axis	min	avg	max
91	activity	2	4.0000	10
91	potency	1	4.2632	10
91	evaluation	2	5.2105	10
102	activity	2	4.8947	10
102	potency	2	4.4737	10
102	evaluation	2	6.3684	10
104	activity	4	6.5263	9
104	potency	2	4.8947	8
104	evaluation	3	6.7368	9
107	activity	5	6.8947	9
107	potency	1	5.0526	9
107	evaluation	2	6.1579	9
116	activity	3	5.7895	9
116	potency	3	5.1053	9
116	evaluation	4	6.1053	9
126	activity	3	5.7368	8
126	potency	2	5.4211	8
126	evaluation	3	5.2105	9
130	activity	2	4.9211	10
130	potency	1	5.9211	10
130	evaluation	2	5.7105	10
132	activity	3	5.6842	8
132	potency	2	5.3158	8
132	evaluation	3	5.7368	8
136	activity	3	6.7895	10
136	potency	3	5.3158	9
136	evaluation	3	6.3684	9

Table 7.17: Ratings based on the tempo

Instrumentation

tempo	axis	min	avg	max
activity	2	4.8772	10	
potency	1	5.1053	10	
evaluation	2	5.5789	10	
activity	2	5.8421	10	
potency	1	5.5263	10	
evaluation	2	5.6667	10	
activity	3	6.2895	10	
potency	3	5.2105	9	
evaluation	3	6.2368	9	
activity	4	6.5263	9	
potency	2	4.8947	8	
evaluation	3	6.7368	9	
activity	2	4.8947	10	
potency	2	4.4737	10	
evaluation	2	6.3684	10	

Table 7.18: Ratings based on the number of instruments used

tempo	axis	min	avg	max
1	1	2	5.3596	10
1	2	1	5.0526	10
1	3	2	5.8684	10
2	1	2	6.0000	10
2	2	2	5.3421	9
2	3	3	6.0263	10

Table 7.19: Ratings based on the mode

Mode

7.2 Mappings

Mappings

Both mappings, the exemplary mapping and the anti-mapping can be downloaded at [**?**]

Bibliography

[1] Rudolf Arnheim. *New essays on the psychology of art*. University of California Pr, 1986.

[2] David Brang and VS Ramachandran. Survival of the synesthesia gene: why do people hear colors and taste words? *PLoS biology*, 9(11):e1001205, 2011.

[3] Richard E Cytowic. *Synesthesia: A union of the senses*. The MIT Press, 2002.

[4] Clyde L Hardin and Luisa Maffi. *Color categories in thought and language*. Cambridge University Press, 1997.

[5] Alexander Hauer. Physics-based music visualization, 2013.

[6] Dick Higgins and Hannah Higgins. Intermedia. *Leonardo*, 34(1):49–54, 2001.

[7] Johannes Itten. *Kunst der Farbe*. Reinhold Pub. Corp, 1961.

[8] Kaarle Nordenstreng. A comparison between the semantic differential and similarity analysis in the measurement of musical experience. *Scandinavian Journal of Psychology*, 9(1):89–96, 1968.

[9] Charles E Osgood. Semantic differential technique in the comparative study of cultures1. *American Anthropologist*, 66(3):171–200, 1964.

[10] Stephen E Palmer, Karen B Schloss, Zoe Xu, and Lilia R Prado-León. Music–color associations are mediated by emotion. *Proceedings of the National Academy of Sciences*, 110(22):8836–8841, 2013.

[11] Andreas Schmid. Physics-based music visualization, 2013.

[12] Karen Schloss Stephen E Palmer. An ecological valence theory of human color preference. *PNAS*, 2010.

[13] Zoe Xu Lilia R Prado-Leephen E Palmer, Karen B Schloss. The measurement of meaning. *Urbana*, 1957.

[14] Zoe Xu Lilia R Prado-Leephen E Palmer, Karen B Schloss. New essays on the psychology of art. *Berkeley, CA: University of California Press*, 1986.

[15] Sabine Susstrunk, Robert Buckley, and Steve Swen. Standard rgb color spaces. In *Color and Imaging Conference*, volume 1999, pages 127–134. Society for Imaging Science and Technology, 1999.

[16] Tawny Tsang and Karen B Schloss. Associations between color and music are mediated by emotion and influenced by tempo. *The Yale Review of Undergraduate Research in Psychology*, page 82, 2010.

[17] Johann Wolfgang von Goethe. *Zur Farbenlehre*, volume 40. Jazzybee Verlag, 1963.