Creating and evaluating a particle system for music visualization

Joyce Horn Fonteles a,b, Maria Andréia Formico Rodrigues c,*, Victor Emanuel Dias Basso a,b

a Centro de Ciências Tecnológicas (CCT), Universidade de Fortaleza (UNIFOR), Avenida Washington Soares 1321, J(30), 60811-905 Fortaleza, CE, Brazil
b Centro de Humanidades, Universidade Estadual do Ceará (UECE), Avenida Paranjana 1700, 60714-903 Fortaleza, CE, Brazil
c Programa de Pós-Graduação em Informática Aplicada (PPGIA), Universidade de Fortaleza (UNIFOR), Avenida Washington Soares 1321, J(30), 60811-905 Fortaleza, CE, Brazil

Article history:
Received 8 February 2013
Received in revised form 20 September 2013
Accepted 6 October 2013
Available online 15 October 2013

Keywords:
Particle system
Music
Visualization
Animation
Evaluation

Abstract

In this paper, we present a simplified 3D particle system and fast translation algorithm we have designed and implemented to generate real-time animated particle emitter fountains choreographed by a classical music. The approach we used to translate and map the controlling information into the musical fountain animation is also introduced, as well as the process to build the physical model of the music fountains. A proof of concept is implemented to demonstrate the main system’s aspects, its feasibility, and that it has met the system’s design goals. Moreover, it shows that is possible to observe visual patterns that match the theme of the musical composition, as an example of how the system can be used not only for visual appreciation and entertainment, but also as a possible support tool for music composition. We have also conducted a user study as an evaluation of the system. The results of this have provided us with positive and useful feedback on the effectiveness of our visual mappings as well as further directions to explore.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Virtually, many of the traditional ways of representing and analyzing music have been automated, in different degrees, by computing methods, techniques and tools [1]. Due to the fact that music evolves in time, the use of computational technologies for the production and analysis of music has become more common between computing researchers and composers. Actually, the musical science has been advancing in the construction of innovative systems using distributed computing, artificial intelligence, evolutive computing, virtual reality and interfaces for gestual music control [2].

With the advances in computer graphics, we can now rapidly generate representations of sound and music [3,4]. Several types of geometric patterns have been used to visualize musical elements, forms and structure: arc diagrams [5], isochords [6], bars [7,8], circular forms [9], random fractals [10], spirals [11], cylinders [12], three-dimensional objects [13,14], etc. For example, a lot of creative work has been done over the years using 3D particle systems. One of the most common uses of them is to create animated visual effects such as clouds, smoke, liquids and explosions [15].

In fact, visualization is one of the most promising approaches to aid the learning curve of music audience, particularly, the general audience, in understanding the sophisticated structure in classical music, thanks to our
strong visual cognition ability [3,16]. Although many efforts have been done to the visualization of the sonic characteristics, only few attempts have been reported to visualize the structure of the music. The challenges of visualizing the music structure lie in both pre-processing of the input data and the design of a reasonable graphical representation for music. Also, the potential of visualizing musical structure and elements in 3D space and of validation based on users studies has not been fully explored yet.

This work explores some possibilities of producing a 3D animation as a visual extension of a musical performance, through the mapping of color, movement, space, and time to represent the relationship between the visual patterns and the music. In particular, as music can be described as highly organized sounds that exhibit time-varying structures in the pitch and time domains [17], the visualization of these time-varying pitch contexts forms the musical theme is one of the main goals of this work. To accomplish this goal, we have designed and implemented a simplified 3D particle system and fast translation algorithm to generate real-time animated particle emitter fountains choreographed by music. The approach used to translate and map the controlling information into the musical fountain animation is also presented. A proof of concept is implemented using the Symphony No. 5 by Beethoven to demonstrate the main system’s aspects, its feasibility, and that it has met the system’s design goals. Moreover, it shows that is possible to observe visual patterns that match the theme of the musical composition, as an example of how the system can be used not only for visual appreciation and entertainment, but also as a possible support tool for music composition. We have also conducted a user study as an evaluation of the system. The results have provided us with positive and useful feedback on the effectiveness of our visual mappings as well as further directions to explore.

2. Related work

A variety of computer animations that stimulate the imagination to the momentary music have been proposed in response to the strength, tempo, rhythm, pitch, etc., with interesting results. Kubelka, for example, describes a prototype application for music visualization that consists of three main units: sound analyzer, visualization module, and scene editor [18]. Similarly to our work, a particle system is used to model a fountain object to express the dynamics and mood of the music. However, the visualizer displays few perceptual characteristics of digital audio. Besides, results of proof-of-concept testing to demonstrate and evaluate the feasibility of the application are not presented. In particular, the mood is a difficult characteristic to map and, more importantly, to evaluate. The analysis of the musical expression aimed by an interpreter, for example, makes possible to predict the emotions and the musical expressiveness of a piece. The data extracted can be mapped into colors and shapes that can be used, for example, to create a mechanism for the visualization of a musical expressiveness. This kind of visualization is becoming an interesting pedagogic tool for students of music and the use of algorithms for the extraction of musical data has become relatively common, even for polyphonic music [19].

Among the most common parameters of extraction, we can quote the pitch of a note, the moment in which it is played, and its respective volume [20]. Some works present the extraction of parameters of a musical performance; mathematical models based on time and frequency values for synthesis and musical analysis [21]; and the extraction, in real time, of a musical expression from an audio sound.

Some works focus on how to map color to sound and vice versa [4,6,22–24]. In the sound and music computing domains, analyses of resonant and musical elements have been generally extracted from MIDI (Musical Instrument Digital Interface) files and sound waves [4,8,13,25]. Some of these efforts include the development of computer systems for music visualization that automatically capture and display visual structures as a computer animation that enables people to see, and better understand, the tonal patterns that they hear.

A work described by Smith and Williams seems to have also a number of parallels to ours [26], including the use of OpenGL and MIDI components. More specifically, these authors have developed a program that parsed MIDI data files and generated 3D graphical representations based on the MIDI information. They also used a mapping function to transform the music data into 3D objects. However, the mapping strategies they employed are different from ours as well as the 3D scene modeled, which consists of a small number of spheres, to represent the MIDI data for a piece of music. Other authors [25,27] presented a similar mapping scheme as the one proposed by Smith and Williams [26]. More specifically, Hiraga et al. developed a system where colored cylinders are used to represent single notes, and the different MIDI channels of the piece are stacked along the z-axis [25]. The pitch of a tone is mapped to the y-axis, the volume to the diameter of the cylinder, and the tempo to the cylinder’s color saturation. As the music plays, a scan-plane that is orthogonal to the layers moves from left to right to allow the user to see how the notes in the scene correspond to the music heard. Dickie envisioned the development of a novel application as an environment where users with varied musical backgrounds could create music by sculpting 3D objects that subsequently would have their spatial characteristics mapped to sound parameters [27].

The Max family of applications (Max/MSP/Jitter [28] and PureData [29]) are popular choices for composing interactive digital media works and implements a visual Data Flow Architecture through a flexible patching interface. However, the Max visual data-flow environment presents some inherent limitations [28]. For example, it focus on visual representation of process interconnection (it hinders expressive data structures, dynamic graphs and data-structures are problematic, procedural control flow is difficult, it has minimal variable scoping, etc.); most processor nodes are black boxes (it limits process control granularity, etc.); and block-rate quantization of audio controls (it provides a separate scheduler & semantics for control and audio, sample-accurate granularity and events only within black box). Also, in the commercial version there are several audio and MIDI limitations imposed by a Live authorization. Other
commercial products have also been developed to visualize the structure of music (TimeSketch [30], MusicAnimation Machine [31], Hyperscore [32], etc.). Malinowski has also used MIDI to show that a musical score can be understood in an intuitive way by anyone, without the necessity of years of training and conventional study of music, establishing, in this way, an important connection between the musical static notation of the score and the dynamic movements of a music [33]. Chew and François developed a multimodal interactive system for music analysis and visualization using real-time MIDI input from a live performance, revealing tonal structures such as pitches, chords and keys [11]. Unfortunately, the problem with most of these systems is lack of extensibility, thus each new attempt starts from zero, rather than building on previous work. Besides, user studies in all these papers are not well designed or are simply absent.

3. Overview of the system

3.1. Design and functional requirements

The basic design principle was to create a modular system, avoiding design decisions that would commit the system to specific musical variables, such as number of simultaneous notes played, etc. Besides the advantages of simplicity and generality, the modularity of the design also facilitates, for example, the replacement of system components with alternative implementations.

Therefore, a set of requirements were formulated for our system: the prototype should predict the realization of thorough analyses of sound, the management and handling of large amount of data, and generate computer animations to be shown simultaneously with the music. The choice of a particle system for this representation was resulting from its geometrical simplicity, fluidity, and dynamic of its elements that exhibit a wide range of interesting and rich behavior, since a musical work presents subtle nuances, stylistic interpretations and different structural patterns. Particle systems are 3D representations composed of a set of elementary particles that have a life cycle. Each of these particles has its own attributes like position, color, size, etc., which evolve through time. One of the main advantages of particle systems over traditional techniques is their capacity to react dynamically to a change in parameters or a modification of the environment [34]. Among these parameters, we have defined a particle as a structure which contains: the size of a particle, its color, its trajectory (position and velocity), its age, its mass, its acceleration, its rendering method, and its lifetime [35]. Each of these parameters can be set individually. We have also designed particle emitters (continuous source of particles over time) to create some more elaborate visual effects. In our system, all the particles move under a gravity force. Lifetime is used to limit the total number of particles active during the animation.

3.2. Visual appearance mapping

It is unlikely there is a single best visual mapping for music that suits all purposes and contexts. Considering that classical music has a well-defined structure, which is quite time-varying, we chose to generate a number of particles in particle emitters. This process creates evolving particle streams, which can be displayed interactively by our visualization system. These graphical elements are intuitive and do not interfere with each other when combined to form the complete visualization of a music composition. Additionally, using different sizes of particles can lead to elaborate visual effects that are not very expensive in comparison with other rendering techniques. Each emitter acts as one of the possible sources of the particles, and has attached to it a set of particle behavior parameters. A time-based scale changes the visual properties of the particle during its lifetime, for example, its color, an essential aspect of how we see the world.

The first idea of relating the paintings and, consequently, their associated colors to the musical notes is not new. Actually, it dated of the century XVIII, when Isaac Newton associated each of the seven colors that he saw in prismatic light with the seven notes of the harmonic scale [36]. Other artists like Castel, Bainbridge and Rimington created their own mappings between colors and notes [37]. The idea of relating the notes of the musical scale to various colors happened many times over the centuries. As described by Collopy [36], beyond their obvious arbitrariness, these scales do not account for how differently we make sense of visual and audio information: a pattern that sounds harmonious does not necessarily look harmonious.

Differently from other works, the current implementation of our system employs a color mapping developed by Louis Bertrand Castel in 1734 [38], as shown in the second row of Fig. 1. Castel’s model was chosen due to the fact it establishes different colors for each one of twelve musical existent notes making colors transient the way that musical notes are, for example, differing from Newton’s model, displayed in the first row of Fig. 1, which does not consider accidents of semi-tones, such as flattened and sharpened notes.

Each note was then associated, in accordance with Castel’s studies, with a particular color that would be displayed in the position of each individual particle upon the playing of each note (Table 1).

3.3. Implementation

Our 3D particle system prototype was developed using a Java programming language binding for the OpenGL 3D graphics API. Java has been widely used in the development
of musical applications due to its portability and facility for learning, and comes with the JavaSound libraries. The Java Sound API is a low-level API for effecting and controlling input and output of audio media. It provides not only explicit control over the capabilities commonly required for audio input and output in a framework that promotes extensibility and flexibility, it also generates applications which generally do not use much CPU. A MIDI data file is then taken as input. Basically, the Java Sound API can be used for controlling audio playback, audio capture, MIDI synthesis, and basic MIDI sequencing. Besides, Java jMusic API [39] can playback directly using these libraries. It provides a solid framework for computer-assisted composition in Java, and is also used for generative music, instrument building, interactive performance, and music analysis. In Fig. 2, the overall layout of the implemented system is shown. In the beginning of the animation frame, new particles are produced by the emitter fountains.

The size of a particle is a floating point number that represents the space occupied by it. A color of a particle is a 4-component structure that contains the RGBA (red, green, blue, and alpha) values. We initialize the colors of new particles and update them at each new frame of the animation. The current color of a particle is a linear interpolation between the initial and the final colors, where the initial color is the color at its birth and the final is the color at its death. We define the trajectory (specified as a position and a velocity) of a particle as a collection of successive states, each state depending on the previous one. We assume that we have a constant time interval $\delta t$, between two successive states. The age of a particle and its lifetime are extensively used by many methods in the system (size, color, trajectory, etc.). Thus, they need to be initialized and updated accordingly. We used the age of a particle as a decision variable to set its death. In our particle emitters, we defined a constant emission rate (an integer number of particles launched in a certain time interval) to simulate a stream of particles. For each particle emitter, given the number of particles $n$ to emit per second and the time interval $\delta t$ between the current frame and the next one, the number of particles to emit for this frame is $n \times \delta t$. However, $\delta t$ is a decimal number, whereas $n$ is an integer. So, inspired by [34] we first calculate the decimal number of particles to emit and add this number to a floating point accumulator, which keeps trace of all the decimal parts of particles to be emitted. Thus, the number of particles to emit in the current frame is the integer part of the accumulator, and the remaining accumulator is the decimal part (which will be used in the next frame).

A range that represents the lifetime sets the minimum and maximum length of time in seconds each particle

<table>
<thead>
<tr>
<th>Note</th>
<th>Color</th>
<th>Note</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Blue</td>
<td>F# or Gb</td>
<td>Orange</td>
</tr>
<tr>
<td>C# or Db</td>
<td>Blue-Green</td>
<td>G</td>
<td>Red</td>
</tr>
<tr>
<td>D</td>
<td>Green</td>
<td>G# or Ab</td>
<td>Carmine</td>
</tr>
<tr>
<td>D# or Eb</td>
<td>Olive Green</td>
<td>A</td>
<td>Violet</td>
</tr>
<tr>
<td>E</td>
<td>Yellow</td>
<td>A# or Bb</td>
<td>BlueViolet</td>
</tr>
<tr>
<td>F</td>
<td>Yellow-Orange</td>
<td>B</td>
<td>Indigo Blue</td>
</tr>
</tbody>
</table>

Table 1
Castel's model used in our prototype for each musical note.
Euler's approximation is responsible for the visualization of the system. Methods that access the graphics library, which are analyzed. In our prototype, the renderer object defines the existent mapping between them and the musical variables. Animation frame and the next one to be generated. In case of accuracy of results.

And low computational cost, while maintaining good expense of more computation. Despite its limitations, the accuracy of results.

In the current version of the system, the traditional Euler’s method is used to solve the ODE that defines the position of a particle in the next instant of time. This numerical method produces an approximate solution for the problem, which consists in moving one step towards the direction of the derivative. For example, the positions and velocities of the particles are functions of one variable, the time. Euler’s algorithm generalizes to systems of ODEs by expressing them in the vector form

\[
\begin{align*}
\frac{d}{dt} x(t) &= \{\dot{x}_1(t), \dot{x}_2(t), \ldots\} \\
&= \{f_1(x_1, x_2, \ldots, t), f_2(x_1, x_2, \ldots, t), \ldots\} = f(x(t), t)
\end{align*}
\]

where \(x(t)\) is the solution vector and the right hand side \(f(x, t)\) is the derivative vector. Then the vector form of Euler’s approximation is

\[
x(t + \delta t) \approx x(t) + f(x(t), t)\delta t
\]

The smaller we make the step length \(\delta t\), the more accurate the approximation. However, this comes at the expense of more computation. Despite its limitations, the choice of Euler’s Method can be justified by its simplicity and low computational cost, while maintaining good accuracy of results.

The state of the system is updated for the next instant of time, given by the time step between the current animation frame and the next one to be generated. In case of the visual attributes, they are directly controlled by the existent mapping between them and the musical variables analyzed. In our prototype, the renderer object defines the methods that access the graphics library, which are responsible for the visualization of the system.

### 3.4. Extraction of musical characteristics through MIDI files

Inspired by the idea of Malinowski [33], we built a prototype system using an accurate MIDI file that illustrates a variation of the traditional musical notation to be used by people with or without any formal music training. We believe that MIDI files can be accepted as digital score input on which well-established music analysis and algorithms can be employed, providing the users with some initial results. Besides, each message from a MIDI file contains a series of already established musical parameters, thus avoiding the need of processing the majority of required data [42]. In particular, we extract three variables when a note is played: tonal pitch, volume and timbre. In our particle system, these three music terms are assigned to the following 3D computer graphics attributes:

1. Tonal pitch determines which color will be used to paint each particle (dark and light colors have effects which are comparable to low and high musical tones, respectively; dark colors are sonorous and powerful, while light colors are floating and light) [43];
2. Volume is represented by the increase of the size of the particles; and
3. Timbre characterizes which instrument emitted a specific musical note, being represented by a 3D emitter of particles.

To capture the sent messages, it was also necessary to establish the lines of entry and exit of MIDI data, creating a synthesizer in charge of effectively playing the notes. Each MIDI event that reaches the object sequencer is analyzed: in case of a Note-on, the three variables are extracted, mapped and sent to the renderer so that it updates the particle visual attributes; and in case of a Note-off the animation is terminated, since the sound ceases of existing (Fig. 3a). The messages of Note-on and Note-off are followed by 2 data bytes: the first one informs the tonal pitch of the note; the second one informs its volume. Velocity is the term we used to define the intensity of the touch: if the touch is strong, the velocity is high, consequently, the sound is very intense; however, if the touch is weak, the velocity is low, in other words, the sound is not much intense (Fig. 3b).

Since the data entry is a pre-existent file, our system extracts all the MIDI data from the events Note-on and Note-off, in a sequential list that includes: the timestamp, a sequence of characters denoting the time at which a certain event occurred, of every note played; its respective channel, through the attribute channel; its tone height, that is determined by the attribute pitch; and the volume, represented by the attribute velocity, that indicates the strength used to play that note [20]. From these data, it is necessary to capture a message that indicates the end of the music, represented by a control event called MetaEvent.

The next step is to map each event of note played to its respective visual features. The channel whose note is played establishes which emitter is responsible for the emission of particles that will correspond to that specific sound.

When there is only one instrument playing in a musical piece, there are no problems to execute it through MIDI,
however, in cases where several instruments are playing simultaneously, it is necessary to establish a different channel (altogether 16 channels) for each instrument. There is a standard for the distribution of the instruments in the channels [44].

Regarding the volume of a sound, other two types of messages need attention: Aftertouch or Channel Pressure, and Key Aftertouch or Polyphonic Channel Pressure; since they can determine alterations in this and in other sound attributes, like resonance, vibrato, etc.

The last type of MIDI message considered in this work is the Pitch Bend or Bender, which represents an alteration of tuning in a note. This message can be recognized by almost all the musical electronic instruments, except the electronic battery (because it has not tonal height). Since the human ear is very sensitive to alterations of tuning, the message Pitch Bend is followed by two bytes of data, being the first byte more significant than the second one.

4. Proof of concept

The most common way to learn music is through studying a musical score, which contains the objective notations of a music composition. However, beginners have to spend a considerable amount of time to learn the basics of music theory before being capable to understand the notations and to mastering a musical score and, particularly, a classical one. Moreover, untrained people may only be able to feel the sound elements (pitch, rhythm, volume and speed); while people who have received extensive training in music theory and history usually know the actual structure and form of the music [3,16]. Therefore, for our proof of concept we chose to exploit the high learning curve of a very well known classical music.

4.1. Theme

The theme, particularly in the Western Music, is the main concept developed in the course of a piece of music. It is typically integrated to several passages of the music and is able to vary in its extension. In our case study, the piece of music chosen was the first movement of Beethoven’s 5th Symphony transcribed for string quartet (Fig. 4). We focus on the first movement since it is commonly regarded as the signature of a symphony. More specifically, this motif works like a unifying structure during the whole piece of music, so much in the rhythmical like melodic form, being repeated constantly, transposed (i.e., a collection of notes is moved up or down in pitch by a constant interval), and executed by different instruments. In this manner, it defines the fundamental parameters or elements of the piece.

4.2. Visual modeling

The selection of visual attributes and their appropriate use must translate the dynamic quality that the chosen music offers. Therefore, it is important to create a visual modeling that allows the user to become focused and involved with the visual component of the performance.

More specifically, we modeled four emitters of particles to represent a string quartet, the most popular form of chamber music, composed by two violins, one viola and one cello. These musical instruments are positioned like they would be in an orchestra, where the sharpest, highest in pitch, instruments are on the left and the ones that play low notes are on the right, as shown in Fig. 5. This representation aims at establishing a straight relation with the chamber music.

We generated then the corresponding computer animation that is composed of 472 frames. Since each note has a corresponding timestamp, it was possible to use this information in the synchronization of keyframes of the animation. However, for doing this it was first necessary to convert ticks of a MIDI event to frames of the animation. Inspired by the solution proposed by Bain [45], we use an alignment between the visualized frames and the MIDI ticks, a critic factor for the synchronization between music.
and visualization, in the form to satisfy our objective of creation of data for the keyframes.

4.3. Testings and results

Preliminary testings show, by comparing some musical passages and the corresponding computer generated animations, that the participants in general were able to identify frames in which all the emitters have the same colors, indicating that the instruments played a note in unison (left side of Fig. 6), and frames where the emitters present a clear individual behavior, with different colors and particle generation moments, representing the musical figures of tonal pitch, rhythm and pauses (right side of Fig. 6).

Besides, it is also possible to identify frames where two or more emitters spread out particles with a set of equal colors, however, in a reverse position (like the one displayed on the left side of Fig. 7), indicating that the 2nd violin and the viola played reverse melodic movements. The middle diagram of Fig. 7 shows that the emitters spreading out particles with more colors represent instruments that executed, more quickly, a greater number of notes. And the diagram on the right side of Fig. 7, showing the biggest particles, clearly indicates that the sound played was more intense, with a stronger volume.

In practice, melody might change or it might repeat throughout a song or piece. So, the use of particle emitters arranged like a string quartet to illustrate these ongoing changes in the Symphony No. 5 by Beethoven, along each particle behavior and the animation time line, was able to translate in a simple visual form, a portion of the history told by the music, particularly for those viewers with little classic musical knowledge.

The presence of the theme is then easily perceived through the visual patterns that are established, for example, illustrated at the moments in which the particles begin and stop being produced, in their alterations of colors, as well as in the changes of size that they suffer. These patterns repeat every time we listen to them. It helps us to establish the central idea of the piece and to notice what is the musical way followed by it, through an instrument, and consequently through its corresponding emitter of particles that executes these patterns at a given moment. In particular, we perceive that the visual patterns that auto-repeat unify the computer animation like the theme unifies the musical piece.

5. Evaluation

Our system for music visualization was evaluated by general audience, music students, and music experts. We performed user studies to quantitatively evaluate the proposed visualization with special attention to the participants’ different levels of experience and, particularly, to the general audience’s opinion and reaction. User surveys were also conducted with all participants, this time, with special attention to the group of music students and experts to collect their qualitative feedback on the visualization.

5.1. Method

One computer animation was created for use in this study, the same one described in Section 4. Thirty three participants with a variety of musical experience were selected. They were divided into three groups: Group 1, Group 2, and Group 3. More specifically, Group 1 was formed by seven players of our University Camerata, a group of expert musicians all of whom have 10–20 years of classical music education and practical experience. These players show a natural, very detailed and quick response to music visualization. Group 2 was composed of thirteen very young undergraduate music students (aged 17–18), with instrumental playing ability. They reported having a
background in music fundamentals and, on average, three years of ability to play an instrument. Group 3 was represented by a generic audience formed by thirteen undergraduate and graduate students from our University, with listening habits only.

Each participant worked individually to minimize distractions. Initially, they were asked to read the instructions from the instructions sheet and to fill out their profile information. The test proceeded by requesting them to watch the animation exhibited on a screen (while listening to the music using stereo ear plugs) and complete the accompanying form. After viewing the computer animation, the participants were also invited to add any supplementary comments and suggestions they had on the music visualization using a user survey form.

5.2. Results

Tables 2–9 are summaries of responses to several questions asked in the user study. Participants (see rows 1, 2, 3 and Groups 1, 2, 3 of Table 2) found the mapping of tonal pitch, volume, and timbre evident. They perceived timbre (68% of the viewers) and volume (68% of the viewers) but they recognized tonal pitch (75% of the viewers) better. As regards the different colors chosen for mapping the notes when the chords (simultaneous notes) are played, some viewers reported that they found them particularly interesting. Although most people felt they could recognize volume, one participant suggested in the user survey form that the changes in the particles size could be bigger than they are.

The majority of the viewers, approximately 85% (Table 3) accurately identified frames in which all the emitters have the same colors, meaning that the instruments played notes in unison or octave; and 70% of them perceived that the emitters spreading out particles with more colors represent instruments that executed a greater number of notes (Table 4).

Predominantly, the participants recognized the motif in the animation by music rhythm, one of the most important elements in music, with a quite high degree of certainty (around 85%, as shown in the first and third rows of Table 5). Most of them recognized the motif by melodic patterns too. This result was much more pronounced in

---

**Fig. 7.** Comparison of three musical passages and their corresponding computer animation frames.
expert musicians (around 86%) and undergraduate music students (100%) compared to the general audience with listening habits only (around 61%).

Not surprisingly, the repetition of the motif being played was the structure perceived by around 85% of the viewers, as shown in the first row of Table 6. More than half of them (around 58%) recognized the variations of such motif in its transposed form, as displayed in the second row of Table 6, and a little less than half of them (around 45%) perceived it in its retrograde form, as shown in the third row of Table 6. Actually, the retrograde is relatively harder to identify, although it is simpler in theory. Even expert musicians showed a certain difficulty in identifying a retrograde motif in the animation. This difficulty experienced by most participants was probably caused by the fact that this visual pattern was displayed on the screen too quickly to be perceived.

The predominant opinion in our user studies about the capacity of the computer animation to express the mood of the music was positive (approximately 61% of the viewers, as shown in Table 7). However, expert musicians and music students stressed in their user survey forms that interpreting the mood of the music just by visualizing small colored particles displayed simultaneously on a screen is challenging. Even though, because they consider color as a property of visual experience, they also said that the variations in colors used in the animation can inspire emotions. Many of them described that each pigment has a set of feelings that tend to emerge when a color is recognized by their visual system. Some suggested to vary the particles velocity to increase the expressiveness of the animation. As we stated before, the mood is indeed a difficult characteristic to map and evaluate.

As was expected, although approximately 56% of the viewers found that the exhibited animated patterns may be used to distinguish musical genres, most of them found difficult to track these patterns during the animation (Table 8).

Nearly all viewers (more precisely, 97%) agreed that this type of visual experience could become an additional and interesting support for music learning (as shown in Table 9). In particular, the great majority of viewers remarked that a rich visual experience makes easy the understanding of a subject. In addition, they made supplementary comments stating that, actually, a score is for the musician a “visual experience” which helps in the reproduction of the music. Furthermore, others commented that a computer animation also makes it possible to represent visual patterns with an awakening effect to maintain user’s concentration. Many viewers found the animation allows better perception of rhythms and melodies too, besides the association of these musical elements with colors and movements of computer animated characters.

5.3. Discussion

Initial user studies demonstrated a strong perceptual relation between music terms (tonal pitch, volume, and timbre) and visual attributes (colors, size, and emitter), respectively. In addition, many viewers found the animation allows better perception of rhythms and melodies too, as well as the association of these basic elements of music with colors and movements of computer animated characters. Also, the predominant users opinion about the capacity of the animation to express the mood of the music was positive. Many of them described that each pigment has a set of feelings that tend to emerge when a color is recognized by their visual system. Others

<table>
<thead>
<tr>
<th>Table 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can you recognize in the animation the occurrence of the following visual patterns? (a) The repetition of the motif by one or more instruments? (b) The motif transposition? (c) The variation of the motif in retrograde?</td>
</tr>
<tr>
<td>Group 1</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>The repetition of the motif by one or more instruments</td>
</tr>
<tr>
<td>The motif transposition</td>
</tr>
<tr>
<td>The variation of the motif in retrograde</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you believe that the computer animation is able to express the “mood” of the music (happy, sad, lyric, aggressive, etc.)? Order them using grades ranging from 1 (lowest belief) to 5 (highest belief).</td>
</tr>
<tr>
<td>Group 1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you believe that the animated visual patterns of this animation may be used to distinguish musical genres? Use grades ranging from 1 (lowest belief) to 5 (highest belief).</td>
</tr>
<tr>
<td>Group 1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do you think that this type of visual experience can become an additional and interesting support for music learning? To conclude, please feel free to include in the user survey form any comments and suggestions you would like to make.</td>
</tr>
<tr>
<td>Group 1</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>No</td>
</tr>
</tbody>
</table>
suggested the importance to vary the particles velocity to increase the expressiveness of the animation.

The results were therefore encouraging and provided positive feedback as to the usefulness of each mapping. Furthermore, thirty-two viewers agreed that this type of visual experience can become an additional and interesting support, particularly for music theory learning, that could be useful to strengthen and amplify the information processed through the sound input for that the student of music has already been trained. Among the participants, the viewers with no musical experience reported that music visualization can help untrained people who do not necessarily acquire musical knowledge to learn a piece of classical musical work briefly without a score. Participants also suggested that people can benefit from this additional support for music learning because it may help the mind to be alert, concentrated, and active. Most importantly, they said that visual experience can give people diversity and the freedom to rewind and repeat as much as they like, or as many times as they need.

6. Conclusions and future work

A 3D particle system to generate real-time animated particle emitter fountains choreographed by music has been presented. Besides its simplicity, the system includes some important and useful characteristics for interdisciplinary learning, offering students in music and computer graphics new visually oriented experiences and opportunities that may be challenging and at the same time enjoyable. Our system was evaluated by general audience, music students, and experts and showed promising and positive results. The animated particle emitter fountains choreographed by the first movement of Beethoven's 5th Symphony transcribed for string quartet provided a good alternative to the typical music literature essays for understanding a piece of musical work perceptually with minimal expertise. The results of this have provided us with positive and useful feedback on the effectiveness of our visual mapping.

A number of topics can be considered as future work. For example, other classical compositions might be visualized and evaluated using our particle system. Additionally, other also relevant aspects may be explored and incorporated into our system. For example, we have assumed four emitters to represent a string quartet. This was a reasonable assumption for a real time visualization system, but it may not scale well with the size of the ensemble. Besides, it is not a priori obvious that each instrument is mapped to a fountain, therefore we can further improve our system by including other visual styles to visualize sounds. This may lead to another challenging extension of this work that includes the implementation of hardware optimizations. The incorporation of such extensions can lead to expanded knowledge and understanding of music visualization as well as suggesting new limitations and solutions. Finally, we believe that visual art and music are essential in expanding the horizons of people (especially of children) beyond the everyday and thus, introducing the music theory learning in the form of computing games and visual effects would be also a great and useful way for entertainment and of learning about visual language.

Acknowledgments

Maria Andréia Formico Rodrigues is supported by the Brazilian Agency CNPq (The National Council for Scientific and Technological Development) under grant No. 310434/2010-6 and would like to thank for its financial support. Joyce Horn Fonteles has benefited from a scholarship under grant No. 504682/2007-5 and would like also to thank CNPq. Finally, we are particularly grateful to the Editor-in-Chief and referees for providing insightful comments and suggestions to significantly improve the manuscript.

References


[34] F. Caillete, An Interactive Particle System, University of Manchester, 2002 (MSc. Dissertation).


