

Responsive Visualization for Musical Performance

Robyn Taylor	Pierre Boulanger	Daniel Torres
University of Alberta	University of Alberta	University of Alberta
Edmonton, Alberta, Canada	Edmonton, Alberta, Canada	Edmonton, Alberta, Canada
E-mail: robyn@cs.ualberta.ca	E-mail: pierreb@cs.ualberta.ca	E-mail: dtorres@cs.ualberta.ca

Abstract

We present a framework that facilitates the visualization of live musical performance using virtual and augmented reality technologies. In order to create a framework suitable for developing technologically augmented artistic applications, we have defined our system in a way that is modular and incorporates intuitive development processes when possible. In this paper we present a method of musical feature extraction and provide three examples of music visualization applications that we have developed using our system. Our visualizations illustrate features in live singing and keyboard playing using responsive virtual characters, responsive video imagery, and responsive virtual spaces.

1. Introduction

Each time a song is interpreted, the notes and words may be the same, but the experience is unique. Interaction between a vocalist and her fellow musicians, the musicians and the audience, the energy of the room, and the adrenalin rush of performing all contribute to the serendipitous nature of live music, making each performance preciously ephemeral and distinct.

When an interpretive artist performs a piece, she knows *what* she will be singing, but gives herself freedom to decide in the moment *how* it will be sung. Concert or theatrical productions that accompany live musical performance with pre-recorded visuals lack the flexibility to let the artist manipulate imagery as well as sound. In order to provide artists with a truly responsive and spontaneous audio-visual environment, it is important for the visualization medium to be flexible enough to convey the subtle nuances of the live performance. Each repetition of the visualization experience will be unique, since no two live performances will ever be the same.

While traditionally used for scientific purposes, responsive visualization technologies offer artists an exciting new way to create audio-visual art pieces. Advancements in graphics rendering and computer processing power, the accessibility and ease-of-use provided by visually programmed development environments, and the relative affordability of sophisticated visualization hardware have made responsive technologies available to artists wishing to create audio-visual entertainment.

Examples of existing music visualization artworks created by linking scientific visualization technologies with musical control systems include Jack Ox's *Color Organ* [8] which visualizes harmonic structures and instrument timbres within an immersive CAVE environment, Levin and Lieberman's *Messa di Voce* [6] which augments the vocalizations of live performers in a theatrical setting, and *The Singing Tree* [7] created by Oliver et al. as part of MIT MediaLab's Brain Opera which allows participants to experience audio-visual feedback that responds to the sound of their singing.

Our goals, when creating our system, were to devise a schema for extracting musical feature information from live performance, and to facilitate the mapping of audio data onto visual parameters so as to enable artists to create dynamic audio-visual performances that are controlled in real-time through live musical input.

2. Overview

To augment a live musical performance through responsive visualization, musical feature data within a stream of real-time input must first be parsed into discrete and measurable units. Subsequently, this musical feature data can be mapped to responsive imagery.

Our system separates the musical feature extraction tasks from the visualization mechanisms. The musical feature extraction module runs on a Macintosh G5 system, while visualization engines may run on Windows, Linux, Macintosh or Irix machines. Visualization engines receive information about the musical performance by connecting to the musical extraction module via a network connection implemented using the Virtual Reality Perception Network (VRPN) [12]. This allows musical and visual content to be easily re-mapped in different ways. This modular approach encourages code re-use.

Since the system is intended to be used by teams of artists and developers who are creating technically augmented performable works, this system should be as accessible as possible for users who may have little or no formal training in computer programming. We used visual programming environments whenever possible to create our system modules. These environments allow programs to be designed by drawing connections between system modules in order to visualize the way data flows within the application. These environments are often very intuitive for non-technical users.

In this paper we describe the musical feature extraction process and present three examples of mappings between music and responsive imagery that were generated using our system. In each implementation a different mapping is used to illustrate properties of interpreted music:

- Visualization of the emotional content of a musical piece through the behaviour of a virtual character
- Visualization of a singer's vocal timbre using responsive video
- Visualization of vocal dynamics inside a reactive immersive environment

3. Musical Feature Data Extraction

In order to visualize live music, it is essential that the stream of live music be parsed into discrete parameters. Cycling '74's visual programming development environment Max/MSP [3] is specially designed for sound processing, allowing a programmer to easily manipulate audio and MIDI data. We use Max/MSP to create our Musical Perception Filter Layer, which is illustrated in Figure 1.

Our Musical Perception Filter Layer (see Figure 1) extracts the following parameters from live musical performance:

- Pitch: Vocal pitches are identified both in terms of their raw pitch values and in terms of their scale degrees relative to the tonic of the key signature of the sung melody.
- Loudness: Vocal amplitude is transmitted in dB.
- Timbre: A descriptor of the user's timbre is devised. Open vowels (like /a:/) are differentiated from closed vowels (like /i:/).
- Chord: The chords the user plays on the digital piano are identified.

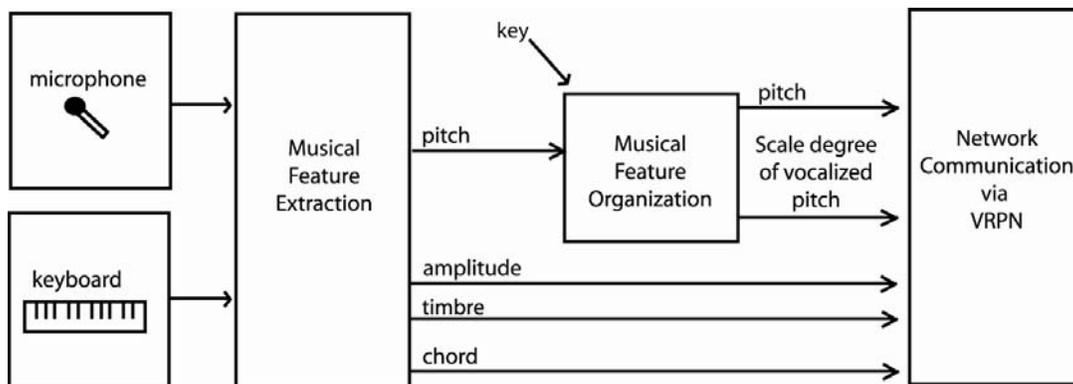


Figure 1: *The Musical Perception Filter Layer*

3.1 Vocal Pitch and Loudness Extraction. Pitch and amplitude information is extracted from sung vocal input using the Max/MSP `fiddle~` object created by Puckette *et al.* [9]. The `fiddle~` object analyzes the incoming sound signal to determine the pitch of the singer's vocalization, using Fourier transformation to convert the signal's complex waveform into a harmonic spectrum. This spectrum contains data describing the frequency and amplitude of each sinusoidal component contained in the incoming sound. The fundamental frequency of the signal is reported as the pitch of the incoming sound, and the signal's amplitude is reported to be the loudness.

3.2 Tonal Encoding of Pitch Data. We use Western tonal music as input melodies to our music visualization systems, so once our raw pitch data is extracted, we then organize it in a way consistent with tonal music theory. Our system organizes vocal input into a tonal context using strategies devised by Deutsch and Feroe [5]. We encode each pitch in terms of its intervallic relationship with the tonic note of the key signature of the melody. This facilitates any music-theoretical analysis we may later wish to do upon the sung input.

3.3 Timbral Descriptors. Since the `fiddle~` object outputs the frequency and amplitude data describing each of the partials forming the harmonic spectrum of the user's singing voice, we can analyze this harmonic spectrum to obtain information about the singer's timbre. Upon examining the harmonic spectrum, we assess the distribution of energy amongst the partials in the sound. If we compare the harmonic spectrum that describes the singer's vocalization to known information about the harmonic spectra characterizing different vowel sounds, we can describe an aspect of the singer's vocal timbre by providing a rough estimation of the singer's vowel choice. The vocalist can manipulate this timbral descriptor by modifying the vowel choice he or she employs while singing.

3.4 Chord Identification. Keyboard input monitoring is done in a separate sub-patch. Our Max/MSP sub-patch monitors MIDI events in order to determine what chords are being played on the keyboard. To do this, we determine which pitches are being played on the keyboard and identify them in terms of pitch class. We consider the note C to have a pitch class of 0, Db to have a pitch class of 1, and so on. We examine the pitch classes that are played on the keyboard and compare them with a list of the pitch classes found in a collection of 'known' major and minor chords. This method permits chords to be played in any inversion. More chord types (diminished, augmented, major and minor sevenths, etc.) could easily be classified and added to the list of known chords, making expansion of this module trivial.

4. Visualizing Music through Virtual Character Behaviour

Our first example of a music visualization application visualizes sung melodies through the responsive behaviours of an artificially intelligent virtual character. In this implementation a character created using Torres and Boulanger's ANIMUS Framework [13] [14] is used to visualize the emotive content of music by expressing simulated emotion through animated behaviour [10][11].

4.1 The ANIMUS Framework. The ANIMUS Framework is used to create believable virtual characters who can respond to events in their environment. Responsive character behaviour is generated using a three-layer process which simulates the perceptual and cognitive processes used to process live musical input and formulate and express an emotional response in real-time:

- **Perception Layer:** The musical features extracted in the Musical Perception Filter Layer are communicated to the virtual character's perception layer via the VRPN connection. In this way, the virtual character is able to perceive salient musical features within the stream of live musical input.
- **Cognition Layer:** In the cognition layer, the musical information obtained in the perception layer is used to influence the virtual character's emotional state. Subtleties of musical phrase and vocal intonation are used to interact with the creature and modify its simulated mood. In the cognition layer, the artistic concept for the link between the nuances of vocal performance and the character's emotive response is defined.
- **Expression Layer:** In the expression layer, the 3D creature's cognitive state is visualized to the audience using dynamically generated animation. The animations are generated by interpolating between keyframe poses that are associated with emotional states.

The ANIMUS Framework is designed with the intention of facilitating artist/scientist collaboration. The system allows artistic designers to create program skeletons outlining how each of the three ANIMUS layers should function. Technical team members then implement the specific functionality in order to create synthetic characters that are capable of communicating believably with an audience of viewers.

4.2 The *Alebrije* Character. *Alebrije* is a lizard-like virtual character (see Figure 2) that was created during the development of the ANIMUS Framework. We extended the *Alebrije* character so that his responses could be used to visualize the emotive content of sung music. The animated imagery is life-sized and can be displayed upon a stereoscopic display, making it suitable for use in augmented theatrical productions where the virtual actors must be consistent in scale with the live performers.



Figure 2: *A singer transitions Alebrije from a neutral to a sad position*

In order to simulate Alebrije's awareness of the emotional signifiers in interpreted music, the research of Deryck Cooke [1] correlating melody and composers' emotional intentions is used to formulate his musical cognition system. Aspects of Cooke's research serve as the basis for Alebrije's cognition processes in this implementation. Cooke associates emotive meaning to the tonal structure of Western melodies, associating each tone in the musical scale to an emotional context. Alebrije's interpretation of the emotional meaning of a melody is consistent with Cooke's metric.

Our Alebrije character is capable of distinguishing 'sad' melodies from 'happy' ones, and displaying responsive behaviours that communicate his simulated emotional state. We are currently exploring the possibility of using musically responsive virtual characters like Alebrije in live performance settings.

5. Visualizing Music through Responsive Video

Our second music visualization example uses responsive video to create a performable multimedia piece that is manipulated by a performer's singing and keyboard playing. Cycling 74's Jitter [2] is a video processing package that can be integrated into the Max/MSP environment. The Jitter package allows users to create responsive video applications by describing (using visual programming) how data flows between Max/MSP and Jitter objects. Jitter can be used for a variety of visualization tasks such as manipulation of video playback, generation of basic 3D animation, or the modification of still images. Using the Jitter environment to colour-edit and layer a selection of video clips, we have created a visualization which illustrates the vocal timbre of a live performer and the harmonic relationships between chords played on a digital keyboard. This visualization system has been used in a live concert setting to perform an interactive piece called *Deep Surrender*.

5.1 Visualizing Musical Parameters. In *Deep Surrender*, the Musical Perception Filter Layer is used to extract feature data describing the vocal timbre of a singer, and the chords played on a digital piano. In this visualization, chords are related to one another with regards to their positions on the music theoretical device, The Circle of Fifths, as was previously explored in Ox's *Color Organ* music visualization environment [8]. The chords that are played affect the colour balance of the video display. To define the relationship between chords and colours, the Circle of Fifths was mapped to the standard colour wheel, making chords that are similar to one another on the Circle similar in colour.

To visualize vocal timbre, the amplitude of the fundamental frequency and second and third partials extracted from a singer's vocalization are mapped to a Red-Green-Blue colour selector. The harmonic spectrum of the user's singing defines the hue that is output by the colour selector. Varying the vowel sound varies the harmonic spectrum of the singing, and therefore outputs different colours. The colours produced by the performer's singing affect the layered videoclips that form the visualization.

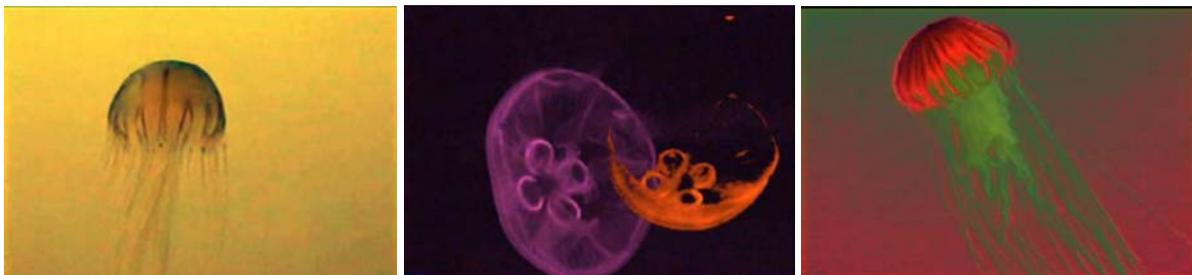


Figure 3: Three phases of the *Deep Surrender* performance

5.2 Deep Surrender. *Deep Surrender* is a multimedia piece written for soprano, synthesizer, and responsive video. The intention of the piece is to illustrate the way an artist can harness anxiety and adrenalin to produce a beautiful performance. This illustration is done through the visual metaphor of a jellyfish – a creature both beautiful and terrifying. The artist's musical performance affects the jellyfish representation, in order to visualize how the artist interacts with and overcomes her anxiety.

In addition to having been performed in concert at the University of Alberta, *Deep Surrender* is routinely performed in the laboratory in order to show visitors how visualization can be used for artistic purposes. It was also performed during several media interviews, including a live performance on CBC Radio.

6. Visualizing Music in Immersive Spaces

Our third music visualization example illustrates vocal dynamic inside a responsive virtual space. The Virtools environment [4] is a visual programming environment which allows designers of virtual reality applications to create immersive visualizations. A musical control system for Virtools applications has been created, connecting extracted musical parameters to behaviours inside the Virtools environment.

6.1 Virtools. Virtools' intuitive authoring environment (see Figure 4) allows different visualization metaphors to be easily defined, tested, and modified. Connecting our musical feature extraction system to the Virtools environment allows us to rapidly develop music visualization applications. This illustrates one of the benefits of our proposed architecture: the connection between Max/MSP's music processing environment and Virtools' virtual reality simulator allows both the musical and visual aspects of music visualization projects to be implemented using visual programming techniques.

6.3 Visualization in Immersive Spaces. The Virtools rendering system is capable of visualizing a virtual environment inside an immersive space consisting of three large screens which display stereoscopic imagery to the users who stand within the enclosure. See Figure 5 for an example of an immersive visualization room. Immersive environments enhance the realism of the virtual experience, as the audience members experience depth perception inside the life-sized space.

6.2 Particle Manipulation through Vocal Dynamics. We have created an example implementation of music visualization within a virtual environment created in Virtools which allows a vocalist to use his or her voice to formulate particle clouds (see Figure 6) generated by Virtools' particle generation and interaction routines. The size and colour of the particle clouds varies in response to the pitch and loudness of the vocalist's singing.

The particle system is one of the built in behaviours that Virtools includes as part of its extensive behaviour library. Virtools authors may not need to manually code any part of the application if the library contains all the behaviours they need. If they do need to custom-create Building Blocks, Virtools makes this possible via a simple SDK that allows developers to code new Behaviors in C++.

Virtools' ease-of-use and extensive Behavior libraries make it a valuable tool in the creation of artistic visualizations. The fact that a visual metaphor can be created and customized in a rapid and intuitive fashion makes it easy to experiment with numerous effects and parameterizations when creating an application. Being able to display the visualizations in an immersive environment increases their expressive and communicative potential.

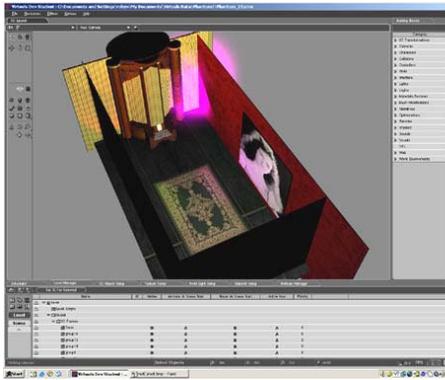


Figure 4: An example of a *Virtools Composition*



Figure 5: A participant inside the University of Alberta's immersive *VizRoom*



Figure 6: A particle cloud triggered by vocalization

7. Discussion

This paper has presented a music visualization system which operates in a distributed fashion, facilitating easy re-use of the musical analysis module contained in the Musical Perception Filter Layer.

The Musical Perception Filter Layer was implemented in the musical development environment Max/MSP [3]. Pitch, amplitude, timbral information and chord data were used to interact in real-time with aspects of a virtual environment.

Three experimental examples were implemented in order to show how the features extracted by the musical analysis module could be mapped to different visualization metaphors:

- Interactive ANIMUS virtual characters [13][14] were used to illustrate the emotive capacities of music [10][11] through visible behaviours that illustrated emotional responses to music. The cognitive system used to trigger these behaviours was consistent with Deryck Cooke's research correlating melody and emotion [1].
- Cycling '74's Jitter environment [2] was used to create responsive video streams which responded to vocal and keyboard input. This visualization was used to create a multimedia performance piece, *Deep Surrender*.
- The Virtools [4] visual programming environment was used to create a visualization which can be performed in an immersive virtual space. This visualization used particle dynamics to illustrate vocal performance.

This framework simplifies the process of generating multiple visualization metaphors to express extracted musical feature data. The virtual character and immersive environment visualizations are currently being used to develop artistic pieces, and the responsive video production has already been used in live performance.

Care has been taken to develop this system in ways that enable rapid development of creative visualizations and collaborative work between artists and programmers. Max/MSP, Jitter, and the Virtools environment can be visually programmed, making possible the rapid prototyping of visual metaphors. The fact that the system allows visualization and musical feature extraction to be conducted in visually programmed environments makes it more accessible to artists who may have no formal training in computer programming. Although the ANIMUS environment does not yet support visual programming,

ANIMUS is designed with the idea of task delegation in mind, allowing artists and programmers to work alongside one another to develop creative works.

We look forward to the continued use of this system for the purpose of creating artistic works that combine live music and responsive visualization.

Acknowledgements

The source video footage for the *Deep Surrender* video production was filmed by Melanie Gall.

The textures on the models used in the Virtools simulation are from <http://www.ktn3d.com/>.

The use of the VRPN library was made possible by the NIH National Research Resource in Molecular Graphics and Microscopy at the University of North Carolina at Chapel Hill, supported by the NIH National Center for Research Resources and the NIH National Institute of Biomedical Imaging and Bioengineering.

References

- [1] Deryck Cooke. *The Language of Music*. New York: Oxford University Press, 1959.
- [2] Cycling '74. Jitter, 2004.
- [3] Cycling '74. Max/MSP, 2004.
- [4] Dassault Systèmes. Virtools, 2005.
- [5] Diana Deutsch and J. Feroe. The internal representation of pitch sequences in tonal music. *Psychological Review*, 88:503-522, 1981.
- [6] Golan Levin and Zachary Lieberman. In-situ speech visualization in real-time interactive installation and performance. In *Proceedings of The 3rd International Symposium on Non-Photorealistic Animation and Rendering*, pages 7-14. ACM Press, 2004.
- [7] William Oliver, John Yu, and Eric Metois. The Singing Tree: design of an interactive musical interface. In *DIS '97: Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 261-264. ACM Press, 1997.
- [8] Jack Ox. 2 performances in the 21st Century Virtual Color Organ. In *Proceedings of the fourth conference on Creativity & Cognition*, pages 20-24. ACM Press, 2002.
- [9] M. Puckette, T. Apel, and D. Zicarelli. Real-time audio analysis tools for Pd and MSP. In *Proceedings of the International Computer Music Conference*, pages 109-112. International Computer Music Association, 1998.
- [10] Robyn Taylor, Pierre Boulanger, and Daniel Torres. Visualizing emotion in musical performance using a virtual character. In *Proceedings of the Fifth International Symposium On Smart Graphics*, pages 13-24. Springer LNCS, 2005.
- [11] Robyn Taylor, Daniel Torres, and Pierre Boulanger. Using music to interact with a virtual character. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 220-223, 2005.
- [12] Russell M. Taylor II, Thomas C. Hudson, Adam Seeger, Hans Weber, Jeffrey Juliano, and Aron T. Helser. VRPN: A device-independent, networktransparent VR peripheral system. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 55-61. ACM Press, 2001.
- [13] Daniel Torres and Pierre Boulanger. The ANIMUS Project: a framework for the creation of interactive creatures in immersed environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 91-99. ACM Press, 2003.
- [14] Daniel Torres and Pierre Boulanger. A perception and selective attention system for synthetic creatures. In *Proceedings of the Third International Symposium On Smart Graphics*, pages 141-150, 2003.