Miró: a flexible expressive audiovisual system for real-time performance & composition

by

Fernando Enrique Franco Lizarazo

A Thesis Submitted for the Degree

Of

Master of Science by Research

Department of Computer Science and Information Systems
University of Limerick
Limerick
Ireland

Supervisor: Dr. Niall J. L. Griffith

Submitted to the University of Limerick, June 2004

Abstract

Miró: a flexible expressive audiovisual system for real-time performance and composition

by

Fernando Enrique Franco Lizarazo

This thesis discusses various issues for designing a flexible expressive audiovisual system for real-time performance and composition, and proposes a design for an audiovisual performance system and discusses its implementation.

I begin by reviewing the principal control metaphors in software-based audiovisual systems (AVS) and then propose a number of desirable factors for building an interactive AVS extracted from this review.

I continue by evaluating various systems in terms of factors for building interactive audiovisual environments. The main issues for flexibility and expressiveness in the generation of dynamic sounds and images are then isolated. Finally, I present a description of the design and implementation of *Miró*, a real-time audiovisual system prototype based on sound and graphics synthesis controlled through a drawing device, where various ideas presented in this document have been implemented.

Acknowledgments

I am deeply indebted to the following individuals for their support and inspiration during the work on this thesis.

Niall Griffith
Mikael Fernström
Liam Bannon
Anne Murphy
Krispin Leydon
Michael Cooke
Golan Levin
Annette Aboulafia
Sergi Jordà

I would like to thank the Pure Data (PD) community for providing technical support through their email lists and websites.

Thanks also to the persons I have had the opportunity to meet in my visits to Media Lab Europe and MIT and at the New Interfaces for Musical Expression (NIME) conferences in Dublin and Hamamatsu. Their projects and ideas have been very important for the definition of this thesis theme and results. These could not be possible without the support of the Interaction Design Centre (IDC) at the University of Limerick.

Thanks finally to my girlfriend Diana and to my family in Colombia for continuous support during this period in Ireland.

Contents

CHAPTER 1. INTRODUCTION **9**

1.1 SYNOPSIS OF THE THESIS 11

CHAPTER 2. BACKGROUND TO INTERACTIVE AUDIOVISUAL SYSTEMS **13**

2.1 CONTROL METAPHORS IN AUDIOVISUAL SYSTEMS 14
  2.1.1 Timelines and Diagrams *14*
  2.1.2 Control-Panel Displays *16*
  2.1.3 Reactive Widgets *18*
  2.1.4 Painterly Interfaces *21*
2.2 FACTORS IN INTERACTIVE AUDIOVISUAL SYSTEMS 25

CHAPTER 3. COMPARATIVE EVALUATION OF EXISTING AUDIOVISUAL SYSTEMS ... **28**

3.1 GOLAN LEVIN'S PAINTERLY SYSTEMS 28
  3.1.1 Yellowtail *29*
  3.1.2 Loom *31*
  3.1.3 Warbo *33*
  3.1.4 Aurora *35*
  3.1.5 Floo *37*
3.2 HYPERSCORE 39
3.3 METASYNTH 42
3.4 VIDEODELIC 44
3.5 MUSIC SKETCHER 45
3.6 FMOL 47
3.7 SUMMARY 49

CHAPTER 4. THE DESIGN OF FLEXIBLE & EXPRESSIVE AUDIOVISUAL SYSTEMS **53**

4.1 INTRODUCTION TO APPROACH 53
4.2 GESTURAL CONTROL 55
4.3 DYNAMIC VISUAL FEEDBACK 62
4.4 SOUND SYNTHESIS AND MUSICAL ASPECTS 65
4.5 MAPPING FLEXIBILITY 68
4.6 SUMMARY 70

CHAPTER 5. DESIGN AND IMPLEMENTATION OF *MIRÓ* **72**

***5.1 MIRÓ* – SYSTEM STRUCTURE** 72
5.2 THE CONTROLLER 74
5.3 SOUND ASPECTS 76
  5.3.1 Sound Synthesis Techniques *77*
    5.3.1.1 Simple Frequency Modulation (FM) 77
    5.3.1.2 Phase Modulation 80
    5.3.1.3 Phase Aligned Formant (PAF) 82
5.4 IMAGE ASPECTS 85
  5.4.1 Graphics Synthesis Techniques *86*
    5.4.1.1 The Paintbrush Tool 87
    5.4.1.2 The Spray Tool 88
    5.4.1.3 The Fountain Tool 91
5.5 SETUP AND CONTROL 93
  5.5.1 Performance Schema *94*
  5.5.2 Composition Schema *98*
5.6 SUMMARY 103

CHAPTER 6. ANALYSIS OF *MIRÓ* **105**

6.1 STRUCTURE AND DESIGN ASPECTS 105
6.2 INTERACTION ASPECTS 106
6.3 EVALUATION 108
  6.3.1 Real-time performance capabilities for the creation of images and sound *108*
  6.3.2 Compositional structures: events organization and modification *108*

6.3.3 Expressiveness ... 109
6.3.4 Mapping flexibility between image and sound ... 109
6.3.5 Modifiers, effects and filtering for audio and image ... 110
6.3.6 Learnability ... 110
6.4 SCENARIOS ... 111
6.4.1 Live Audiovisual Performance ... 111
6.4.2 Audiovisual Composition ... 112
6.4.3 Abstract Animation ... 112
6.4.4 Music Therapy ... 112
6.5 FUTURE WORK ... 113

CHAPTER 7. CONCLUSION ... 115

BIBLIOGRAPHY ... 118

APPENDICES ... 124

APPENDIX A. EXPERIMENTING WITH *MIRÓ*. ... 124
APPENDIX B. *MIRÓ* APPLICATION FEATURES. ... 128
APPENDIX C. STILLS FROM *MIRÓ* IN ACTION ... 130
APPENDIX D. APPLICATION'S CODE. SCREEN SHOTS – PD PATCHES. ... 131
APPENDIX E. CD-ROM CONTAINING THE *MIRÓ* APPLICATION, DEMO MOVIES AND SOUNDS. ... 139

List of Figures

Figure 1: A screenshot from Digidesign's Pro Tools, showing some of the available timeline views for musical information in a typical sequencer and multi-track (Digidesign, 2004). 15
Figure 2: A large Moog system circa 1970, from Moog sales literature. 16
Figure 3: A photo of the Yamaha DX7 synthesizer from 1983. 17
Figure 4: The ReBirth RB-338 by Propellerhead Software (Propellerhead Software, 1999) 17
Figure 5: A screen shot from Pete Rice's Stretchable Music in use (from Levin, 2000). 19
Figure 6: a) Devices used for the VMI environment. The dotted lines represent a 3D virtual surface which the sound/music composer or performer manipulates b) Example of touching the virtual rubber sheet with both hands in Mulder's prototype (from Mulder, 1998). 20
Figure 7: a) Music Insects installed at the Exploratorium b) Screenshot of a SimTunes environment. 23
Figure 8: a) A screenshot of Floo b) A still capture from Aurora c) Screenshot of Yellowtail (from Levin, 2000) 24
Figure 9: A screenshot from Yellowtail, showing its square spectrogram patch in the center, with its horizontal current time indicator (from Levin, 2000). 29
Figure 10: The spectrogram interface patch in Yellowtail (from Levin, 2000). 30
Figure 11: A screenshot of Loom in use (from Levin, 2000). 32
Figure 12: The user's gestures recur periodically in Loom. 32
Figure 13: A screenshot from Warbo. 34
Figure 14: Stills captured from Aurora. 35
Figure 15: The relationship of Aurora's image and audio generators to its underlying particle simulation. 36
Figure 16: a) A screenshot of Floo in use b) Floo's granular synthesizer maps a particle's orientation to the pitch of a Shepard tone in a grain. In this way, the particles can move in a seamlessly circular pitch space (from Levin, 2000). 38
Figure 17: a) Hyperscore screenshot showing four motif windows and one sketch window b) Harmony line (from Farbood et al., 2004). 41
Figure 18: A screenshot of MetaSynth's palettes. Visual patterns (Image, Filter, Wave Table) can be created by using familiar paint tools (UI Software 1998). 43
Figure 19: A screenshot of Videodelic's Input Window (UI Software, 2000). 44
Figure 20: A screen shot of Music Sketcher. 46
Figure 21: A screen shot of FMOL in action. 48
Figure 22: A Digital Musical Instrument representation (from Wanderley, 2001). 54
Figure 23: A flexible & expressive audiovisual system representation. 54
Figure 24: A representation of the mapping distribution. 69
Figure 25: A representation of the mapping strategies. 70
Figure 26: Outline of the performance schema in Miró. 73
Figure 27: An input-output representation of a Miró track. 74
Figure 28: General diagram for simple FM synthesis. 78
Figure 29: a) Block diagram for simple FM synthesis (from Puckette, 2003) b) A screenshot of Miro's simple FM synthesis patch. 79
Figure 30: a) Block diagram of FM synthesis realized as phase modulation (from Puckette, 2003) b) A screenshot of the FM synthesis (phase modulation) patch in PD with a sawtooth (phasor~) and a cosine (osc~) oscillator. 81
Figure 31: Block diagram of the PAF generator (from Puckette, 2003) 83
Figure 32: Calculation of the time-varying parameters, a (the waveshaping index), k, and q for use in the block diagram of Figure 31 (from Puckette, 2003). 83
Figure 33: A screenshot of the PAF generator implementation in Miró. 84
Figure 34: Miró's Paintbrush synthesis algorithm. 87
Figure 35: A screenshot of diverse visual forms created with the Paintbrush tool. 88
Figure 36: An image generated with the Spray tool. 89
Figure 37: Miró's Spray tool synthesis algorithm. 90
Figure 38: Images generated with the Fountain tool. 91
Figure 39: Miró's Fountain tool synthesis algorithm. 92
Figure 40: A screenshot of Miró's main and tracks control panels. 93
Figure 41: Rehearsal and record set up controls. 95

Figure 42: A screenshot of Miró's colour control panel. *96*
Figure 43: Mapping switches matrix in Miró. *96*
Figure 44: Three different visual representations of the same gesture a) Paintbrush b) Spray c) Fountain. See mappings in Table 10. .... *97*
Figure 45: Other visual results from the same gesture in Figure 44 using a different mapping. *97*
Figure 46: Radio buttons (and mapping box) for switching the graphics tool in the graphics control section. *98*
Figure 47: Timelines view in Miró. *99*
Figure 48: Controls for creating and modifying loops in Miró. *100*
Figure 49: An example of a polyrhythm (2 against 3) built with 2 monorhythm objects in PD. *101*
Figure 50: A screen shot of Miró's rhythm control panel. *102*
Figure 51: A screenshot of Miró in action. *102*
Figure 52: Two sequences generated with Miró. *130*
Figure 53: General control module. *131*
Figure 54: Input module. *132*
Figure 55: Graphics window control module. *132*
Figure 56: Gesture time module. The important object is timer. *133*
Figure 57: Track control module. *133*
Figure 58: List manager module, the core of this module is the msgfile object. *134*
Figure 59: Closer view of the list manager module. *134*
Figure 60: Mapping switcher module. *135*
Figure 61: Graphics and Audio Synthesis switcher module. *135*
Figure 62: Master audio level module. *136*
Figure 63: Audio Output module. *136*
Figure 64: Loop module. *136*
Figure 65: Timelines module. *137*
Figure 66: Rhythm patterns module. The core of the module is the monorhythm object. *138*

List of Tables

Table 1: Mappings between image properties and sound parameters used in the Loom synthesizer (from Levin, 2000). 33
Table 2: Comparison of different AV systems. 49
Table 3: Controllers and techniques for generating image and sound. 52
Table 4: Miro's FM synthesis mapping example. 80
Table 5: Miro's FM synthesis mapping example. 81
Table 6: A mapping example stylus-graphics-PAF (sound) synthesis. 85
Table 7: A mapping example Stylus/tablet – Spray Tool - PAF (sound) synthesis. 89
Table 8: A mapping example Stylus/tablet – Fountain Tool - PAF (sound) synthesis. 93
Table 9: Control and synthesis parameters in Miró. 96
Table 10: Mappings between gesture and graphics for the visual representations shown in Figure 44. 97
Table 11: Mappings (changes in bold) between gesture and graphics for the visual representations shown in Figure 45. 98
Table 12: Mappings between gesture and graphics used for the tests. 124
Table 13: Mappings between gesture and sound used for the tests. 124

Chapter 1. Introduction

“For notwithstanding man’s historic efforts to bridge the two worlds of music and art through dance and theatre, the computer is his first instrument that can integrate and manipulate image and sound in a way that is as valid for visual, as it is for aural, perception”.
John Whitney (1976).

The use of visual representations to specify music has a long history. The advent of new interfaces for computer systems offers the opportunity to go beyond the traditional static timeline representation of the score.

The use of flexible technologies such as real-time graphics and sound synthesis facilitates the creation of variety of audiovisual products. Gestural control is also useful, to add expressiveness to these results by taking advantage of a rich and improvised human input, which is important for achieving lively products in computer systems. Through gesture the audiovisual output may be directly controlled by the performer/composer.

An interesting idea in audiovisual systems is to blend the activities of composing and performing. The flexibility that an audiovisual system offers for the user to reconfigure the relationships between image and sound, and control them at various levels, is a key issue in supporting creativity and the realisation of compositional ideas.

Most existing Audio-Visual Systems (AVS) (e.g. *FMOL, Hyperscore, Cubase etc*) emphasize either the graphics display or the sounds for mapping flexibility and control. Therefore, the composer/performer has to adjust to what a system’s designer has decided beforehand about how the form and characteristics of images represent the identity and quality of sounds and vice versa.

Another interesting aspect of various existing audiovisual environments is the use of *dynamic image* as a representation of the sound. In this way there are two dimensions closely linked in these systems: space and time. Some of these systems take advantage of the implicit temporality and spatiality of gestural ‘mark-making’ (drawing, painting) to add expressiveness to the visual and sonic outputs. Other systems that offer visual dynamism are based on predefined visual objects that can be modified and animated.

However, there is an important challenge in the visual representation of the evolution of events in time when you are dealing with a *dynamic image* (animation). In terms of composition, if there are several animated audiovisual sequences or events a *global view,* in which a common time reference is implemented and relevant spatio-temporal characteristics of the sequences are represented, is useful for having a better understanding and control over the temporal relationships between individual sequences.

Most of the existing systems that have adopted the use of *dynamic image* as a representation and specification of sound, lack such a global (temporal) view. This is understandable in the context of real-time performance systems that do not have editing capabilities. I believe this is an important issue that is open to improvement in order to have more control over the composition process with real-time audiovisual systems. However, one could ask: Is it worth editing a real-time performance of improvised (experimental) music? How far can the level of sophistication and flexibility of a real-time performance audiovisual system be taken? This issue is directly relevant to the actuation of sound within performance. Specification, whether it be represented in Tibetan neumes or western notation always involves an interpretive gap that is to filled by the performer. Thus the definition of an interface involves crucially how the determinate and expressive aspects of an instrument are to be realized.

Golan Levin’s work on painterly interfaces for audiovisual performance systems and his *Audiovisual Environments Suite* (Levin, 2000) have been an important source of inspiration for this thesis. The work developed on gestural control within the growing community of researchers and designers in the context of *New Interfaces for Musical Expression- NIME* (Camurri et al., 2002; Cook, 2001; Jordà, 2000; Mulder, 1998; Wanderley, 2001; Winkler, 1998) has also motivated and influenced the work presented in this thesis.

Some material presented in this thesis has been previously published in (Franco et al., 2004).

1.1 Synopsis of the thesis

In Chapter 2, *Background to Interactive Audio-Visual Systems* (AVS)*,* I review the principal *Control metaphors in AVS* in software-based editors and then I propose a number of desirable *Factors for building an Interactive AVS* extracted from this review.

In Chapter 3, *Comparative Evaluation of Existing AVS,* I discuss some of the most interesting interactive audiovisual environments taking into account a variety of requirements and their strengths and weaknesses. These include: *Real-time (improvisatory) performance capabilities for the creation of images and sound, Compositional structures*, *Expressiveness, Mapping flexibility between image and sound, Modifiers for audio and image, and Learnability.*

In Chapter 4, *The Design of Flexible and Expressive AVS,* I discuss the main issues that I have isolated through evaluating various audiovisual environments and a review of related research for flexibility and expressiveness in the generation of software-based sounds and images including: *gestural control, dynamic visual feedback, sound synthesis, musical aspects and mapping flexibility.* I discuss these issues and their implications for the design of expressive and flexible audiovisual systems using gestural input.

Chapter 5, *Design and Implementation of Miró,* presents the development of a gesturally controlled system for performing and composing synthesised audiovisual pieces. I discuss the design of the system in terms of its *structure, controller, sound and image aspects,* and *setup and control*. A flexible mapping between *gesture, image, and sounds,* the *simultaneous generation of images* *and sounds,* and the use of *dynamic images to control and represent sound* are presented as the main goals of the prototype.

In Chapter 6, *Analysis of Miró,* I analyse the performance of the *Miró* prototype in terms of its *Structure and Design, and Interaction Aspects.* In these sections I discuss its strengths and weaknesses and propose ideas for improving future versions of the system. In the *Evaluation* section, I assess the *Miró* system according to the set of properties proposed in Chapter 2, as desirable for building an interactive audiovisual system. In a section on *scenarios*, I propose some contexts in which *Miró* can be used. The last section presents some ideas for *future work* and further development of the system.

In Chapter 7, I present the *conclusions* of the thesis.

Chapter 2. Background to Interactive Audiovisual Systems

Common Music Notation (CMN) is the standard musical notation system, which came to its fullest expression in the seventeenth-century in Europe. It is still used today by many composers and performers.

Roads (1996) notes that CMN has been bypassed by electronic and electroacoustic composers for several reasons:

> - It is biased toward the pitch and duration of notes.
> - It has few provisions for the representation of timbre and does not represent spatial trajectories. The note concept is a single-event abstraction and does not account for the mutating multi-event sound complexes possible in computer music.
> - It addresses only one level of musical form; it was not designed to represent an overview of high-level musical structure; neither is it possible to look below the level of a note to examine the details of the evolving sound structure.
>
> (Roads, 1996)

Since the advent of computer systems many visual representations of sound and/or music have been developed. *"Because music can be represented in so many different ways, it is not surprising that new types of editors continue to be developed. Indeed it would be hard to imagine a "universal" editor for all aspects or representations of music. A diversity of editors is potentially a healthy situation, since musicians can select editors that suit their particular approach to music making"* (Roads, 1996).

Also, since the 1920s various graphical techniques for generating sound have been developed including photographic notation of musical tones, photoelectric tone generators, and spectogram-based scores among others (Roads, 1996). Roads categorized this principle as *"graphic sound synthesis"* that aims for a visual approach to sound specification. According to Roads the graphic control of digital sound commenced with the experiments of Mathews and Rosler (1969). Many graphically oriented synthesis software systems such as Iannis Xenakis's *UPIC* (the initial version dates from 1977) and the commerciallyavailable *MetaSynth* (UI Software 1998), have been developed since.

In the next sections I will review the principal *Control metaphors in Audiovisual Systems* (AVS) in software-based editors and then I propose a number of *Factors for building an interactive AVS* that emerge from this review.

2.1 Control Metaphors in Audiovisual Systems

There are four principal metaphors for image-sound relationships in visually specified computer music. This classification follows that proposed by Levin master's thesis (Levin, 2000) thus:

- *Timelines and Diagrams* (TD)
- *Control-Panel Displays* (CP)
- *Reactive Widgets* (RW)
- *Painterly Interfaces* (PI)

2.1.1 Timelines and Diagrams

These systems are the most traditional or "transitional". They offer different views of musical information including standard music notation, digitized sound waveforms, and MIDI notes displayed on a "piano roll" among others (e.g. *Digital Performer, Cubase, Protools, Cakewalk*, etc). They are largely digital representations and extensions of traditional notation.

Levin (2000) described this metaphor for sound-image relationships in the computer's screen as follows:

> "Scores are generally two-dimensional *timeline* diagrams that operate by relating the dimension of time, along one axis, to some other dimension of sound, such as pitch or amplitude, on the other. In traditional music notation, there may be several parallel time axes (called staves), which make possible the synchronization of multiple simultaneous instrumentalists. In addition to Western music notation, other common examples of sound-timelines are waveform displays, player-piano scrolls, and spectrograms.
> What these various sorts of timelines and diagrams share is a reliance on a coded language of graphical conventions in order to convey meaning. Once learned, this elaborate system of symbols and visual relationships, refined by

> generations of composers and typesetters, yields a remarkably efficient way of organizing and producing a large quantity and variety of musical events" (Levin, 2000).

Figure 1: A screenshot from Digidesign's *Pro Tools*, showing some of the available timeline views for musical information in a typical sequencer and multi-track (Digidesign, 2004).

Because of the extensive traditional use of written or graphic languages of music notation by musicians around the world, the predominance of score-based systems in the field of visually controlled computer music is a natural outcome. Thus the modern *sequencer "wraps the functionality of a multi-track recording system around a backbone of one or more editable timeline displays"* (Levin, 2000). In many of the existing systems the views of musical information include: standard music notation, digital audio waveforms, piano rolls, event lists, controller envelopes, metrical grids and graphic faders (Figure 1).

These forms of visual language rely on the *"reader's internalization of a set of symbols, signs, or grammars whose origins are as arbitrary as any of those found in spoken language"* (Levin, 2000). In compositional terms systems based on timelines have the advantage of allowing the organization of multiple events in relation to a common time reference. Having a global view of recorded events and/or sequences is vital for organizing and making decisions about how these events are related or sequenced in order to produce and manipulate compositions.

2.1.2 Control-Panel Displays

These systems are essentially software realisations of the hardware boxes. They often mimic the sound controls (knobs, dials, sliders, buttons) afforded by analog synthesizers (e.g. *Reaktor, Audiomulch, ReBirth*, etc).

In modular analog synthesizers manufactured during the 1970s (e.g. *Moog III, Arp 2500, Buchla 200,* etc.), each sound parameter required a separate knob or switch, and each interconnection between sound-processing modules required a patch cord. Therefore, dozens of knobs, sliders, switches and patch cords might be used for a single patch. These synthesizers gave the musician a *'hands-on'* approach to exploring the possibilities of a particular synthesis patch (Roads, 1996).

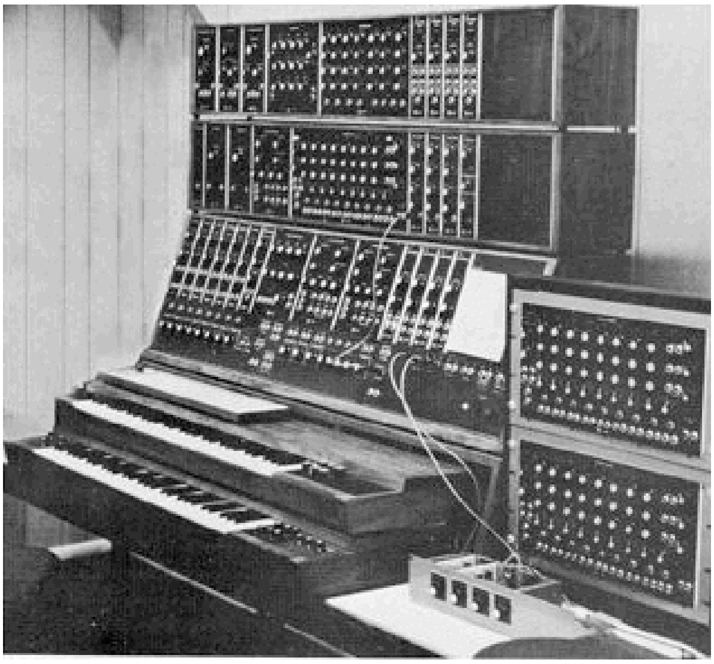


Figure 2: A large Moog system circa 1970, from Moog sales literature.

In the digital domain, many designers have set out to imitate or evoke this sound control pattern. Although digital synthesizers (e.g. Yamaha DX7) offered a wider range of sounds, greater reliability than older analog synthesizers and multiple-function controls, many musicians lamented the loss of the analog knobs they had found so ready-at-hand, expressive and responsive (Levin, 2000).

Figure 3: A photo of the Yamaha DX7 synthesizer from 1983.

In the 1990s the trend of imitating the control panels of analog synthesizers continued in the design of software synthesizers such as Koblo Software's *Vibra6000* and Propellerhead Software's *ReBirth RB-338.*

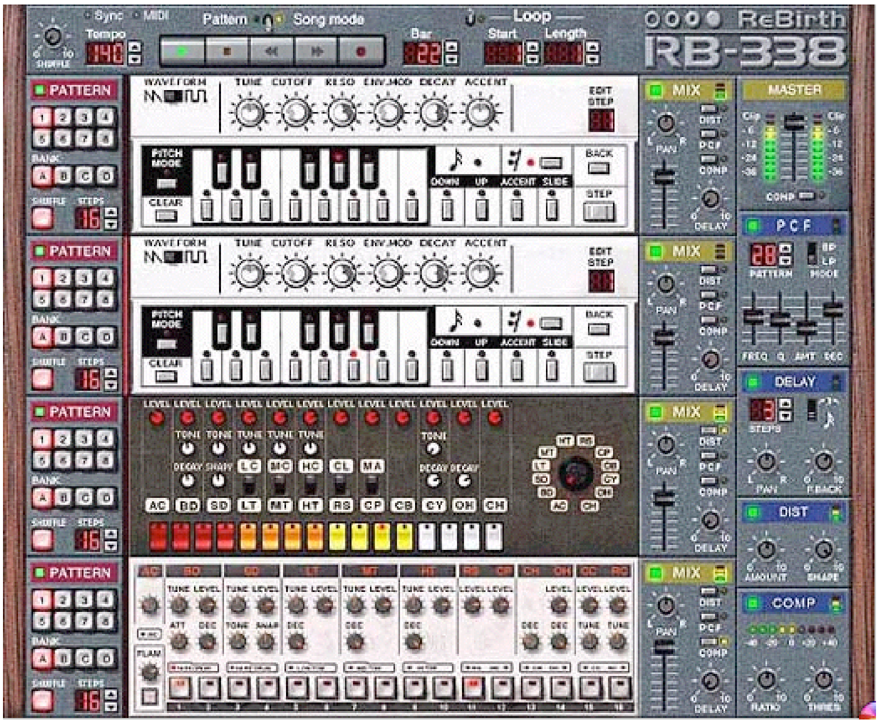


Figure 4: The *ReBirth RB-338* by Propellerhead Software (Propellerhead Software, 1999)

Although these graphic software synthesizers can have a flexible or interchangeable appearance, the use of the control-panel paradigm does have drawbacks such as the confusing homogeneity of a multi-knob interface, and the opaqueness of the mappings from knobs to underlying sound parameters. Also, the user looses the tactile physicality of the original analog synthesizers (Levin, 2000).

2.1.3 Reactive Widgets

These are virtual software objects, which can be manipulated, stretched, etc., by a performer in order to control and modify sounds (e.g. *FMOL, Stretchable Music*, etc.).

Abbado (1998) and Levin (2000) proposed a design approach for screen-based computer music, which *"is built on the metaphor of a group of virtual objects (or "widgets") which can be manipulated, stretched, collided, etc. by a performer in order to shape sounds or compose music. The foundation of this schema is the assumption that a sound can be abstracted as an aural object"* (Abbado 1998, Levin 2000). This approach is analogous to the concept of "slapability" used in DMIX (Oppenheim, 1992).

One of the systems that follow this approach is Pete Rice's *Stretchable Music* (1997) (Figure 5), a system that uses graphical objects for manipulating pre-composed music in real time. *"By using a mouse, the user can stretch and pull animated objects representing different layers of the music. At various times in the system's temporal framework (predetermined by the composer), animated icons pop up on the screen and offer the user an opportunity to play keyboard or drum solos. Users can navigate between four distinct musical sections by grabbing special advancement icons that appear at key points in the piece"* (Rice, 1998).

The use of pre-recorded sound and predefined visual objects in these systems restricts their possibilities of expression and interaction. In some systems more than others, the user has to adjust his/her performance to the arrangement and/or modulation of high-level sonic events (e.g. MIDI sequences) and/or high-level graphic phenomena (e.g. predefined shapes and images) (Levin, 2000). However, some of these systems are more permissive than others. For instance, *FMOL* (Jordà, 2000) permits the generation of sounds from scratch and allows them to be processed widely through the manipulation and/or modification of lines on the screen.

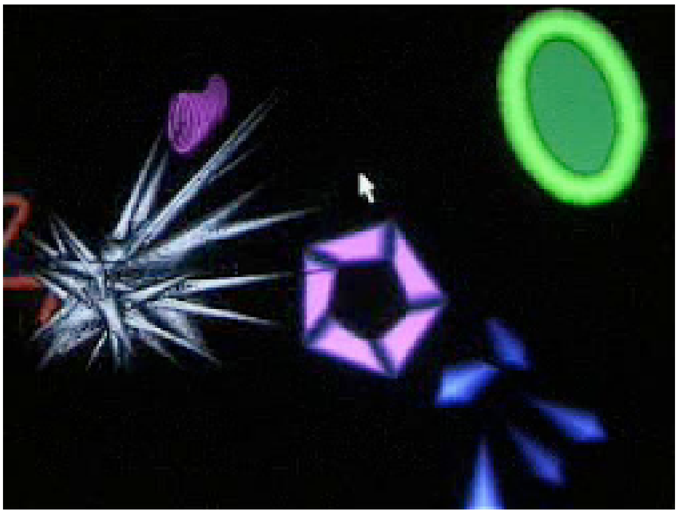
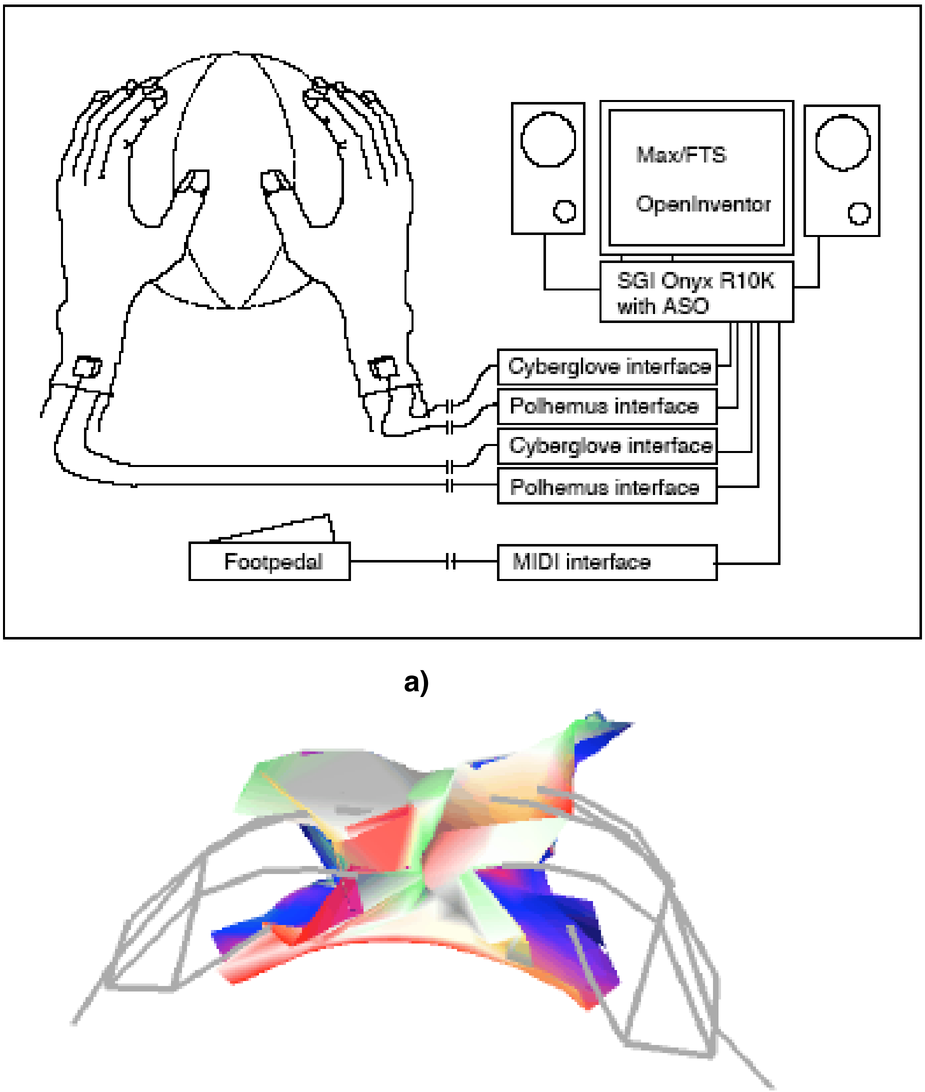


Figure 5: A screen shot from Pete Rice's *Stretchable Music* in use (from Levin, 2000).

Other designers and researchers have created "interactive widget" interfaces that can be gesturally performed and transformed (Jordà, 2003; Mulder, 1998). For instance, in the context of designing virtual musical instruments (VMI), Mulder (1998) developed two prototypes involving virtual input devices with the behaviours of a rubber balloon and a rubber sheet for the control of sound spatialization and timbre parameters. Mulder (1998) described the principles of these prototypes thus:

> "An environment for developing virtual instruments for the control of spatial and timbral sound parameters was created based on the pragmatics of sculpting. In this environment a 3D virtual object, as a visualization of the control surface, is used as input device for the editing of sound - the sound artist literally "sculpts" sounds using a 3D virtual sculpting computer interface. Thus, by changing virtual object parameters such as shape, position and orientation sound and music parameters are changed in real-time. In this environment the object is virtual, i.e. the object can only be perceived through its graphics display and acoustic representations, and has no tactile representation. This environment was implemented using CyberGloves, Polhemus sensors, an SGI Onyx and by extending a real-time, visual programming language called Max/FTS, which was originally designed for sound synthesis. The extension involves software objects that interface the sensors and software objects that compute human movement and virtual object features" (Mulder, 1998).



a)

b)

Figure 6: a) Devices used for the *VMI* environment. The dotted lines represent a 3D virtual surface which the sound/music composer or performer manipulates b) Example of touching the virtual rubber sheet with both hands in Mulder's prototype (from Mulder, 1998).

In Mulder's prototypes virtual object features are used as a means to relate hand movement features to sound features. Some of these prototype's mappings include:

> "Average length of the sheet, along the axis between left and right hand was mapped to flange index. Width (i.e. axis between index and thumb) of the sheet was mapped to chorus depth. A measure of the average curvature was mapped to the frequency modulation index. The angle between left and right edge of the sheet was mapped to vibrato" (Mulder, 1998).

Although this is a highly innovative approach for controlling sound it has the drawback of lacking tactile and force feedback for "touching" the virtual objects, and consequently it is difficult to obtain desired sonic feedback. Also, as stated by the designer, the adaptation of the system to different situations and/or users was not easy without significant technical expertise (Mulder, 1998).

2.1.4 Painterly Interfaces

These systems permit the generation and control of sound and/or musical structures through drawings and free-form images created by making marks in an improvisatory way (e.g. *Music Insects, Meta Synth, RGS, Coagula, Loom, Hyperscore*, etc.).

There is a goal of expressiveness in computer systems based on the painterly metaphor that is similar in principle to that found in the abstract painting movements developed since the early 1910s. These movements include *Der Blaue Reiter* (The Blue Rider) an expressionist art movement founded in 1911 in Germany (e.g. Wassily Kandinsky, Franz Marc), the *Abstract Surrealism* developed in Paris in the mid-1920's (e.g. Joan Miró, Andre Masson), and the *Abstract Expressionism* that flourished in North America from the mid-1940's to the mid-1950's (e.g. Jackson Pollock, Willem de Kooning) (World Book Multimedia Encyclopedia, 2001).

Kandinsky believed that painting, like music, is primarily a form of personal expression, rather than a way to tell a story or state an idea. He created an artistic vocabulary of forms and colors that are expressive but do not refer to anything in the physical world. Kandinsky and the artists who formed the *Blue Rider* movement stressed the spiritual and symbolic properties of both natural and abstract forms.

Abstract painting deliberately omits recognizable subject matter. Instead, the artist explores form, color, design, pattern, and texture to achieve expressive results. Several painting (gestural) techniques have been developed following this approach such as the *action painting* that encapsulates or captures the quickness and energy of de Kooning's brushstrokes and Pollock's “drip” paintings. This philosophy also informed the artistic aims of Joan Miró. *“Miró* *claimed that he generated his forms without preparation or planning, allowing his mind and his hand to wander playfully across the surface of the picture"* (World Book Multimedia Encyclopedia, 2001).

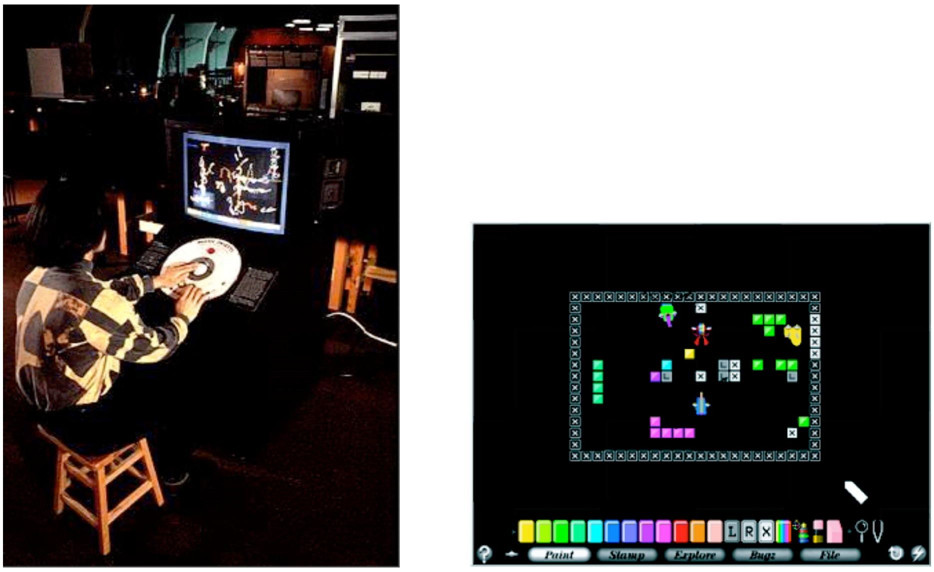
Many artists and music composers have experimented with the correlations of colour/image and music/sound. These include the 19th century composer Nikolai Rimsky-Korsakoff who created a colour coding system for musical tones, and the Russian composer Alexander Scriabin (1872-1915) who experimented with pre-scored musically driven control of lighting and olfactory stimulation in his late works such as *Prometheus* composed in 1911 (Pressing, 1997). This kind of experimentation is related to the concept of *synaesthesia*, which has many different interpretations. For instance, Galeyev (1999) explained this concept as follows:

> «Synaesthesia, at least of that kind which is used in common language and art, is not a "co-sensation" but rather a "co-imagination" or "co-feeling". By its psychological nature it is "association", specifically "intersensory association". And as any association it can be either passive or active, creative kind, having various degree of emotional experience, up to eidetic one» (Galeyev, 1999).

This interpretation of the concept has similarities with the goal of the *Painterly Interface* metaphor in the sense that any visual element can be associated with a correspondent sonic element.

Diverse solutions to meet the goal of *simultaneous authoring of image and sound* in the *Painterly Interfaces* metaphor have been implemented in software-based systems such as Toshio Iwai's *Music Insects* (1991) and Golan Levin's *Audiovisual Environment Suite* (AVES)(2000).

Music Insects, originally designed for the San Francisco Exploratorium, and released as the commercial product *SimTunes* by Maxis Software in 1996, is a visual environment consisting of animated insects that move and react to dots of colour painted by the user on the screen. The system consists of four different insects, each mapped to different instrumental timbres. When a bug passes over a coloured dot, musical scales, sounds, and light patterns are triggered. The insects by default move in a straight line but can be redirected (rotated, reversed) by placing special blocks in their paths (Iwai, 1992). In this way looping rhythms and complex sonic and visual structures can be created. *Music Insects* can be considered as a hybrid, combining aspects of *reactive widgets* (the insects) and *painterly interface* (the coloured dots painted by the user).



a) **b)**

Figure 7: a) *Music Insects* installed at the Exploratorium b) Screenshot of a *SimTunes* environment.

Levin (2000) introduced a new interface paradigm for audiovisual performance instruments based on the idea of an *"inexhaustible, extremely variable, dynamic, audiovisual substance which can be freely 'painted', manipulated and deleted in a free-form, non-diagrammatic context"* (Levin, 2000). Levin described the principles of this scheme thus:

> "According to this scheme, a user creates gestural, painterly marks in a two-dimensional input field, using an electronic drawing device such as a *Wacom* tablet or mouse. These marks are treated as the input to digital signal analysis algorithms, filtering algorithms, and computer simulations. The outputs of these algorithms are then visually interpreted by a graphics synthesizer, and also sonified by an audio synthesizer. Ideally, the mappings which relate the properties of the gestures to their sonifications and visualizations are perceptually motivated, and do not rely on a codified visual or textual language for interpretation. I refer to such a system as "painterly" because I have elected to base its *process* in the act of *mark-making*, in which a gesture is made with respect to some material—as opposed to other domains of gesture, such as sign language or dance—and because part of the *product* of this mark-making

> is, beyond the performance of the mark-making itself, a *two-dimensional image"* (Levin, 2000).

Based on this schema Levin's *Audiovisual Environment Suite* (2000) implemented five new interfaces (*Yellowtail, Warbo, Loom, Aurora, and Floo)* for real-time performance of dynamic visual imagery and sound.

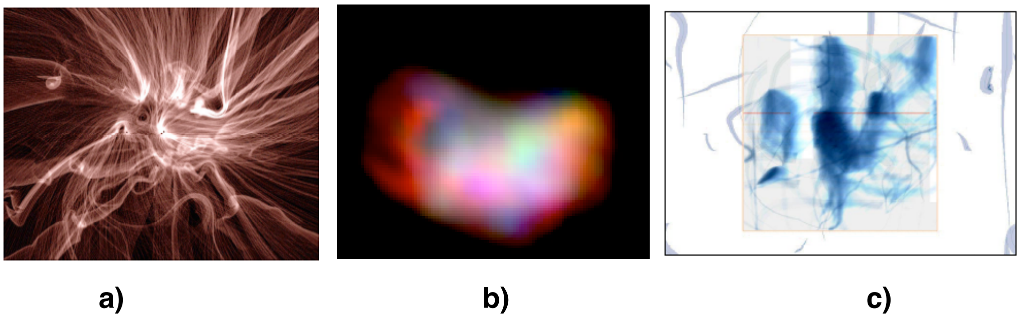

a) **b)** **c)**

Figure 8: a) A screenshot of *Floo* b) A still capture from *Aurora* c) Screenshot of *Yellowtail* (from Levin, 2000)

The diversity of Levin's representational views reflects the desirability of diverse interfaces to reflect the diversity of sonic forms envisaged.

Another important aspect of Iwai's and Levin's painterly systems is the use of *dynamic* images or animations as a result of the user's input. These animations are closely linked to the sound generation and are crucial to produce expressive results.

The four metaphors discussed above broadly reflect two forms of approach. The main difference between the systems based on the metaphors of *Timelines and Diagrams,* and *Control panels,* and on the other hand *Reactive Widgets and Painterly Interfaces,* is that the first group use a visual specification of sounds in which sound and image, especially in the *Control Panels*, are essentially disconnected or independent. In the second group image and sound are more tightly connected. However, while in *Reactive Widgets* the visuals are predefined by the designer and modified by the user to create sonic variations, the goal of *Painterly Interfaces* is the *simultaneous authoring* of image and sound

2.2 Factors in Interactive Audiovisual Systems

From the review of different *Control metaphors in AVS* and various systems built around these approaches, ranging from notation programs to systems that use abstract graphical forms as input parameters for the production of sound or music, I have extracted a number of *Factors for building an interactive AVS* for real-time performance and composition, which can serve as a framework for evaluating some of the existing systems and that I consider useful for the development of my own design. These factors are:

- ***Real-time (improvisatory) performance capabilities for the creation of images and sound****:* This is essential if the aim is to build a system that can take advantage of the human input in an audiovisual environment. The *Painterly Interfaces* approach aims for the creation of *"gestural, painterly marks in a two dimensional field, using an electronic drawing device such as a Wacom tablet or mouse"* (Levin, 2000). This gestural input is connected to graphics and audio synthesizers for creating images and sounds from scratch. An interactive AVS should ideally have these capabilities (Levin, 2000; Abbado, 1998; Rice, 1998; Iwai, 1992).

- ***Compositional structures: events organization and modification****:* Most of the systems that allow the creation of sound and image in real-time don't have the capability for organizing events at a global level. This is however, desirable if the aim is to allow the composition of a piece that involves feedback between sonic and visual events, in the construction of interactive audiovisual compositions. The diversity of global views in systems based on the *Timelines and Diagrams* control metaphor present advantages for organizing and editing events in relation to a common time reference. These properties should ideally be included in an interactive AVS.

- ***Expressiveness****:* An interactive AVS should provide for the *simultaneous* control of sound and image. As discussed before in sections 2.1.3 and 2.1.4, the *Reactive Widgets* and *Painterly Interfaces* metaphors aim for this *simultaneity* of control, which offers advantages for achieving expressive results through different methods. The *detailed control of visual and sonic* parameters is implicit in these approaches. In this way, the use of synthesis techniques for both audio and image is an effective strategy for implementing control at a low-level, and consequently it is useful for adding expressiveness to the system. Also, *dynamic* visual representation linked to the production of sounds as a result of the interaction is a useful aspect for adding expressiveness to an interactive AVS (Jordà, 2003; Levin, 2000; Rice, 1998; Abbado, 1998). Another important aspect for achieving expressiveness in an audiovisual system is the use of gestural input through a physical interface. In this way audio and image can be directly controlled by the performer/composer.

- ***Mapping flexibility between image and sound:*** the variety of approaches to establish image-sound relationships in computer systems, reviewed in section 2.1, indicates that there is no "objective" mapping from sounds to image or vice versa. The diversity of visual representations for different aspects of the music and/or sound is characteristic of the integration found in the *Timelines and Diagrams* control metaphor and is also present in different ways in *Reactive Widgets* systems such as *Stretchable Music* (section 2.1.3) and *Painterly* systems such as *Music Insects* (section 2.1.4). However, in these systems the designer often fixes these correlations between image and sound and the user has to adapt himself/ herself to these design decisions. Therefore, mapping flexibility between the aural and visual dimensions is a desirable feature for the user to feel comfortable with the audiovisual feedback from the gestural input and for expanding the expressive range of the system. This is related to the form of synthesis used. Arguably, some mappings are less arbitrary than others in the case of physical models of synthesis and can be implemented in the system as defaults interaction. However, mappings to spectral synthesis methods are intrinsically arbitrary. The mapping strategies in an interactive AVS can be tackled in many different ways (Rovan et al., 1997; Winkler, 1998; Wanderley; 2001; Mulder, 1998; Ng, 2002; Jordà, 2003; Camurri et al., 2002; Oppenheim, 1992).

- ***Modifiers, effects and filtering for audio and image:*** The systems used in the sound and/or multimedia studio context (e.g. *Protools, Cubase*) offer multiple possibilities for modifying the quality of the recorded material. This is less common in real-time performance systems limiting the variety and/or sophistication of their products. As a part of a complete tool for audiovisual performance and composition, it is useful to implement modifiers, effects and/or filters that can be applied at different levels for both audio and image (Oppenheim, 1992).

- ***Learnability:*** It would be ideal to have a system that is easy to learn and powerful all at once. It is desirable, therefore, to have a system that offers different learning curves. A very cryptic functioning environment would dishearten the user while at the same time he/she will rapidly get bored with one that is too simple (Jordà, 2003; Levin, 2000; Cook, 2000)

Chapter 3Comparative Evaluation of Existing Audiovisual Systems

In this chapter I will discuss some of the most interesting existing interactive audiovisual environments taking into account a variety of properties and their strengths and weaknesses.

I have outlined various control metaphors, and defined a number of desirable properties for building interactive AVS (section 2.2). These include: *Real-time (improvisatory) performance capabilities for the creation of images and sound, Compositional structures*, *Expressiveness, Mapping flexibility between image and sound, Modifiers for audio and image, Learnability.*

The systems discussed in the following sections all make use of one or more of the four principal control metaphors in AVS described in Chapter 2 (**T**imelines and **D**iagrams, **C**ontrol **P**anels, **R**eactive **W**idgets, and **P**ainterly **I**nterfaces), and are either specialized in performance or composition. They all make use of some kind of gestural input and synthetic image for controlling and/or specifying sound. These systems are *Yellowtail* (PI)*, Loom* (PI)*, Warbo* (PI)*, Aurora* (PI)*, Floo* (PI)*, Hyperscore* (PI, TD)*, Metasynth* (PI, TD)*, Videodelic* (RW, TD)*, Music Sketcher* (TD) and *FMOL* (RW)*.* One of the most interesting collections of imagistic gestural interfaces is that created by Golan Levin.

3.1 Golan Levin's Painterly Systems

Golan Levin's approach is interesting because it recognizes the variety of possible interfaces a user may relate to. His systems are designed for real-time and simultaneous performance of dynamic imagery and sound. The painterly metaphor is exemplified by five interactive audiovisual synthesis systems: *Yellowtail, Loom, Warbo, Aurora and Floo.* Video clips and still images of these systems can be found at Levin's *Audiovisual Environment Suite* web page (Levin, 2004) and detailed information about the design in his master's thesis *"Painterly Interfaces for Audiovisual Performance"* (Levin, 2000). Here I will continue the discussion to the relationship between visual input and musical output.

3.1.1 Yellowtail

The author describes *Yellowtail* as *“an interactive software system for the gestural creation and performance of real-time abstract animation that repeats a user's strokes end-over-end, enabling simultaneous specification of a line's shape and quality of movement...each line repeats according to its own period, producing an ever-changing and responsive display of lively, worm-like textures”* (Levin, 2000).

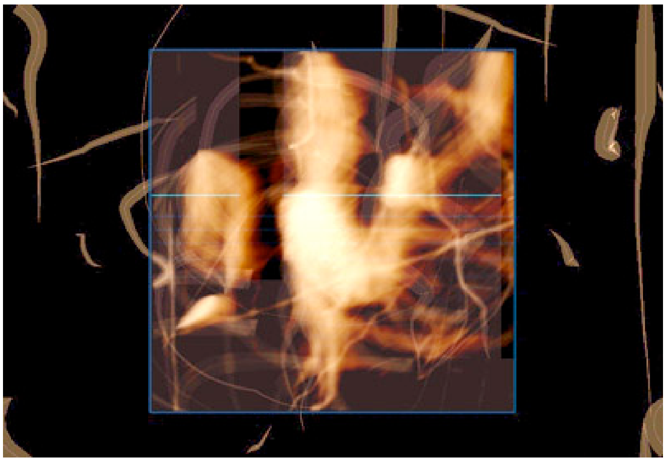


Figure 9: A screenshot from *Yellowtail*, showing its square spectrogram patch in the center, with its horizontal current time indicator (from Levin, 2000).

In order to support real-time sound performance, Levin has provided a square spectrogram patch in the centre of *Yellowtail’s* canvas. The sound is generated by mapping the brightness of pixel columns, in a patch's frame buffer, to the individual amplitudes of a bank of additive synthesis oscillators. *Yellowtail* uses a mapping between sound and image in which change in the *x* dimension maps to pitch and in the *y* dimension to time. A horizontal line called the *current time indicator* sweeps the patch periodically from bottom to top to generate sounds. The functioning of the spectogram patch is described thus:

> “At any given moment this indicator may or may not intersect a row of pixels which belong to one of the user's animating marks. Each of the columns of pixels directs the amplitude of a given sinusoidal oscillator in an additive (Fourier) synthesizer. The greater a pixel's intensity, the more of its corresponding oscillator is heard in the final sound. The oscillators are arranged in order of exponentially increasing pitch from left to right, such that the spectrogram's width spans about six octaves” (Levin, 2000).

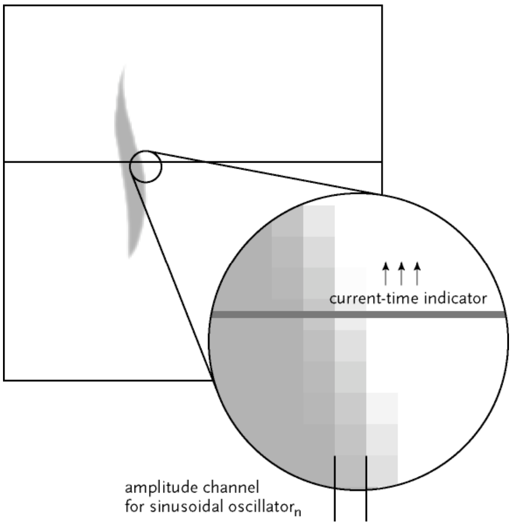


Figure 10: The spectrogram interface patch in *Yellowtail* (from Levin, 2000).

Strengths

- The functioning of the *Yellowtail* system is very easy to learn.
- It permits the creation and performance in real-time of infinite spectrographic image patterns.
- It allows the use of dynamically animated image for playing sounds programmed by the user’s gestures.
- It provides a reasonable amount of control over the spectral content of sound.
- Pitch transposition is easily achieved by moving the spectogram interface.

Weaknesses

- Only a few simple mappings are incorporated between sound and image: amplitude-brightness, pitch–*x* position, and duration–*y* position.
- Only one kind of filter (processor) can be applied in order to modify the recorded sounds and images.
- The mappings between image and sound are inflexible and cannot be changed.
- As the author himself explains: *"the spectogram interface is an extraneous visual element in the image plane and is irrelevant to the user's visual composition, it also segregates the screen into pixels, which make sound, and pixels that don't make sound"* (Levin, 2000). Thus the image is somewhere between being a specification of sound and an integrated element of audiovisual performance.
- From the composition point of view, it does not have editor capabilities. There is no method for reorganizing the events other than to repeat the whole process by making marks in an empty screen.

3.1.2 Loom

In this application Levin's aim was to associate every visual element with a corresponding sound-event. This directness is important, as it allows the user to make sense of what is happening when he/she is interacting with the system using a *Wacom* pen.

According to Levin's description of the system, *"if the user presses harder with her pen, for example, the mark is visually thickened in that location and a louder note is produced at that point in time…at the same time that the user draws the mark, the temporal dynamics of the user's movements are also recorded. As the line is redrawn, its same musical tone is heard, modulated over time in the same manner as when it was first created"* (Levin, 2000).

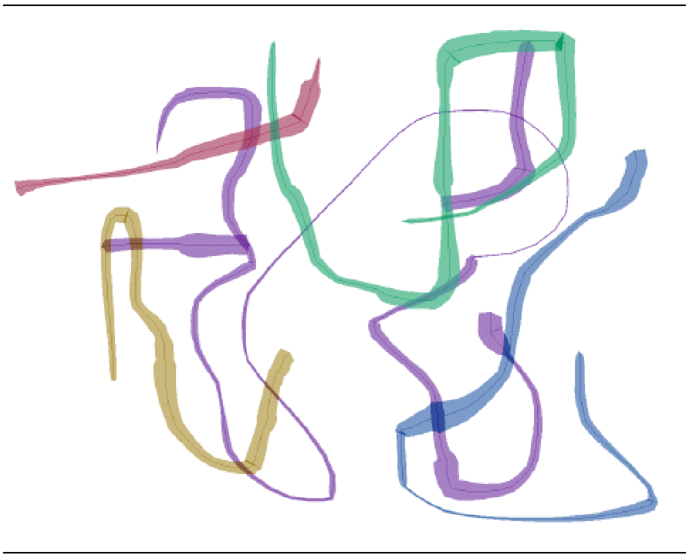


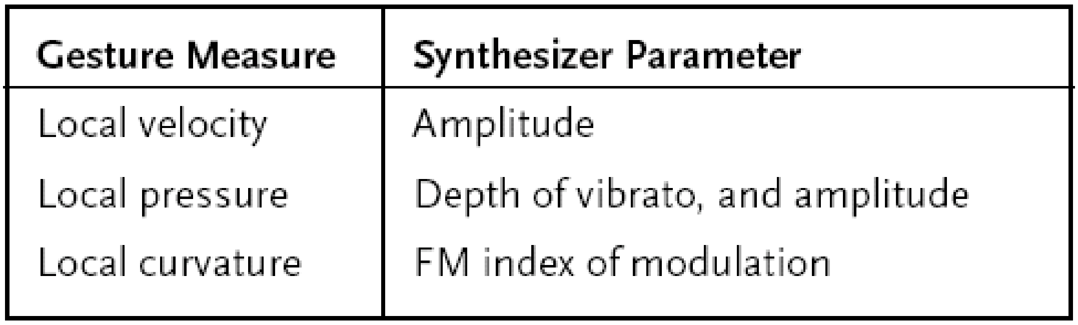
Figure 11: A screenshot of *Loom* in use (from Levin, 2000).

Each mark's playback is synchronized to a common clock whose period is established by the user. This makes it possible to perform marks in a way that produces rhythmic patterns.

Figure 12: "The user's gestures recur periodically in *Loom*. In this example, the lower two marks have the same period, while the uppermost mark has a period exactly twice as long. Although the marks recur in lock-step with each other, an important feature of the *Loom* environment is that each mark can have its own, independent phase in the common rhythmic cycle" (from Levin, 2000).

As Levin explains, the sonification method applied in the system *"is based on the idea that a score or timeline can be wrapped around a user's mark"* (Levin, 2000).

The data contained in a *Loom* timeline includes position, velocity, pen pressure, and local curvature, taken over its length (and hence duration). This data is used to drive the continuous control parameters of a Frequency Modulation (FM) synthesizer.



Gesture Measure	Synthesizer Parameter
Local velocity	Amplitude
Local pressure	Depth of vibrato, and amplitude
Local curvature	FM index of modulation

Table 1: Mappings between image properties and sound parameters used in the *Loom* synthesizer (from Levin, 2000).

Strengths

- Like *Yellowtail,* the functioning of *Loom* is very easy to learn.
- Every visual element is associated with a corresponding sound-event.
- The concept of “wrapping” timelines around the visual marks is very useful in mapping different aspects of the user’s gestures to visual and sonic parameters.
- There is a common clock to synchronize the marks playback. In this way rhythmic textures and patterns can be produced.
- It permits the use of dynamically animated images for playing sounds programmed by the user’s gestures.

Weaknesses

- Certain important parameters of the FM synthesis equation (e.g. carrier and modulator frequencies) are not assigned to properties of the gestural mark (e.g. pressure), and therefore are not directly controllable.
- The system lacks flexible mappings between image and sound.
- It does not support processors that can be applied to modify the recorded sound and images.
- There is no way for reorganizing the events in relation to a global timeline or view.

3.1.3 Warbo

Warbo allows the user to create animated compositions of *“glowing blobs”.* Levin describes the functioning of the system thus:

> "In *Warbo*, the user creates a group of coloured animated spots, each of which corresponds to a pure sine tone. A two-handed interface, which combines the use of a mouse and a *Wacom* tablet, then allows users to control how these tones are made audible. In this way the mouse-hand controls the current pitch(es) and volume(s), while the pen-hand controls the timbre. The user can then create an animated spot, of which there are two possible styles: circular spots, or polygonal spots" (Levin, 2000).

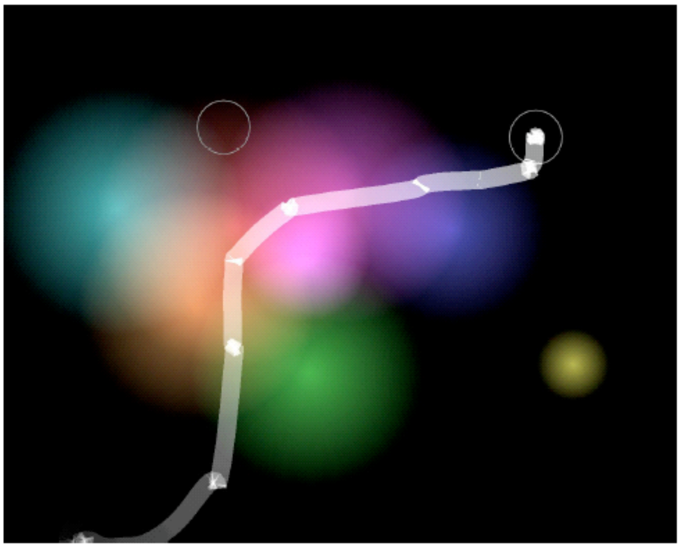


Figure 13: A screenshot from *Warbo*. The user has placed a number of colored spots on the screen, each of which corresponds to a certain sine wave. When the mouse-cursor passes over the spots, a chord is produced whose mixture is based on the position of the cursor in relation to the spots. Meanwhile, a *Wacom* pen in the user's other hand controls a line whose shape governs the timbral content of the chord (from Levin, 2000).

In order to produce tones and chords a cursor is played over the spots, which control parameters from a Chebyshev waveshaping synthesizer.

Strengths

- Each hand can control different aspects of the audiovisual expression.
- The mapping of colour-pitch is a very direct and effective mechanism to relate graphics and sounds.
- The system includes the possibility to generate chords.

Weaknesses

- The mappings between gestures and the sound and image synthesis used in *Warbo* are not very clear for a naive user, and therefore are not easy to learn.

- Again, *Warbo* lacks flexibility in mapping between image and sound.
- There is no way for reorganizing the events along a global timeline.
- Effects and/or filters cannot be applied in order to modify the audio and graphics.

3.1.4 Aurora

According to Levin (2000) *"Aurora is a reactive system whose structural underpainting is a floccular simulation, but whose visual display consists instead of a blurry, shimmering, nebulous cloud".*

Levin explains the visual functioning of this system thus:

> "Superimposed on the terrain of the floccular simulation is a coarse grid of equally invisible square bins. Whenever the display is refreshed, each bin counts the number of floccular particles that occupy it, and assigns itself a brightness proportional to its contents. The bin cells are then visualized using a grid of smoothly-shaded quadrilaterals, which interpolate (with some added hysteresis) their neighbors' brightnesses into their own. By binning and low-pass filtering the simulation in this way, the thousands of data points in the floccular filaments are visually synopsized into an amorphous cloud. Colour variations in *Aurora* are achieved by displaying each filament's density map with a different colour" (Levin, 2000).

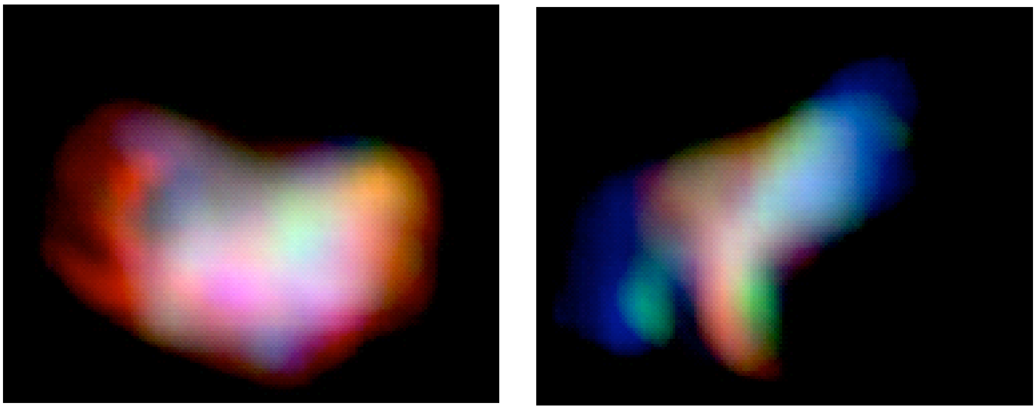


Figure 14: Stills captured from *Aurora*.

Levin selected granular synthesis as *Aurora*'s sonification technique *"because it produces large-scale sonic events in an analogous way to that in which Aurora's graphic system generates macroscopic visual effects: by aggregating the individual contributions of a myriad of infinitesimal atoms"* (Levin, 2000).

To bridge the two domains (visual and auditory), Levin adopted the statistical distribution as an intermediate representation: *"Although this representation had the unfortunate effect of collapsing the dimensionality of both the simulation's information and the synthesizer's control parameters, it worked remarkably well at translating the subtle dynamics of one domain into the behaviour of the other"* (Levin, 2000).

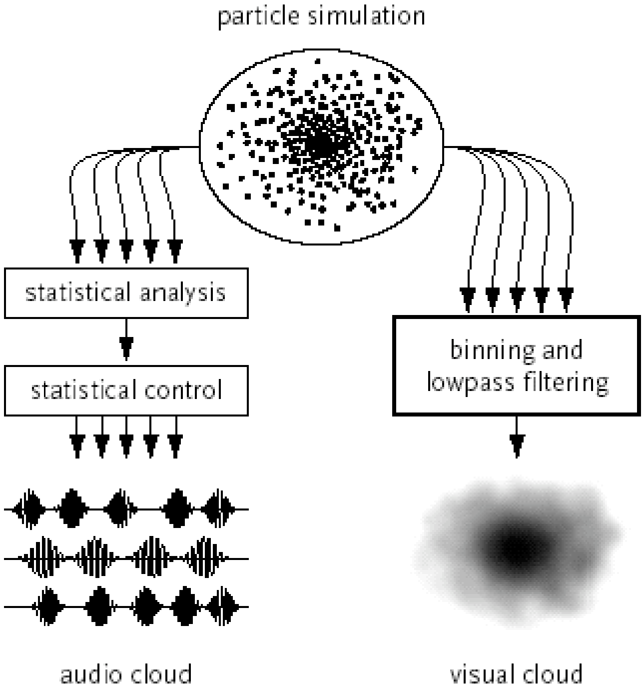


Figure 15: The relationship of *Aurora*'s image and audio generators to its underlying particle simulation. The functional couplet of [*statistical analysis*] and [*statistical control*] is used to conform the dynamic behaviour of the audio to that of the visual simulation (from Levin, 2000).

Strengths

- *Aurora's* functionality is easy to learn.
- It has real-time (improvisatory) performance capabilities.
- It presents an innovative and appealing visual representation.
- Many visual parameters are mapped to the granular synthesizer.

Weaknesses

- The statistical specification and control over the sonic and visual parameters is vague. Most of the products of the interaction are determined by the system.

- The system lacks flexible mapping between image and sound.
- There are no processors that can be applied in order to modify the sound and/or the image.
- There is no way for reorganizing the events in relation to a global timeline.

3.1.5 Floo

Floo presents another interesting approach to the visual and sonic representation of the user's gestures, based on granular synthesis for the sonification and the dispersion and deflection of *"soft-edged tendrils"* in a fluid flow-like simulation for the visualization.

Levin (2000) explains the performance of *Floo* thus:

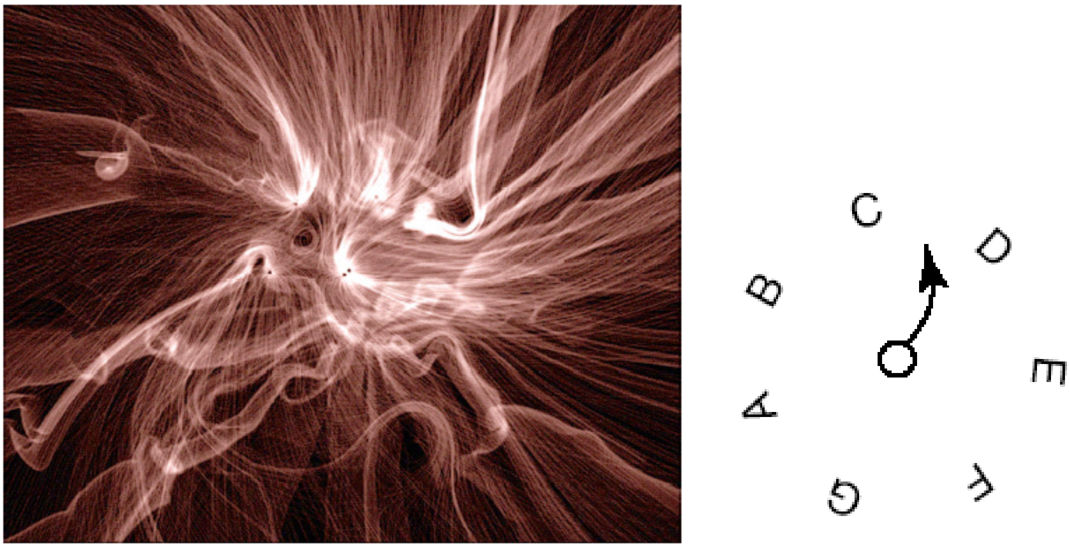
> "Users create synthetic sound and image by depositing a series of fluid singularities (sources and vortices) across the terrain of the screen, and then steering a large quantity of particles through the flow field established by these singularities. An image is gradually built up from the luminescent trails left by the particles; at the same time, sound is generated by a granular synthesizer whose parameters are governed by the dynamic properties of these particles.…*Floo*'s granular synthesizer maps the *orientation* of a particle's velocity to the pitch of a *Shepard tone* [Moore, 1990] [Risset, 1985] used in a stream of grains. Thus, the sonified particles move in a seamlessly circular pitch space. Particles which move in opposite or different directions will create chords, while particles which move in similar directions will create thick, chorused drones" (Levin, 2000).

The expressiveness of the system is based on the blend of *"discrete clicking that creates new sets of particles and spatial configurations of fluid flow, while continuous cursor movements guide particles in real-time"* (Levin, 2000).

Strengths

- It presents an interaction that combines the effects of discrete and continuous gestures.

  The novel use of Shepard tones to fill its grain envelopes mapped to a visual dynamic representation.



a) **b)**

Figure 16: a) A screenshot of *Floo* in use b) Floo's granular synthesizer maps a particle's orientation to the pitch of a Shepard tone in a grain. In this way, the particles can move in a seamlessly circular pitch space (from Levin, 2000).

Weaknesses

- It is not easy to learn because it uses a very subtle and sensitive method to create images.
- Most of the products of the interaction are determined by the system.
- It lacks of a global level for organizing and modifying events.
- It does not support detailed gestural specification and control over the sonic parameters.
- The system is lacking in mapping flexibility between image and sound.
- There are no processors or modifiers that can be applied to the sound and/or the image.

The diversity of Levin's representational views is associated with similarly diverse forms of synthesis and envisaged sound qualities. Strokes, clouds, blobs, tendrils are transformed into sonic analogies; linear, diffuse and discrete sonic events. The diversity of views reflects the desirability of interfaces that reflect the diversity of sonic forms envisaged. However, although Levin has created a diversity of representations, this diversity and/or flexibility is across a set of discrete separate systems. They are not present within a unified system.

3.2 Hyperscore

This prototype developed by Farbood and Pasztor (2001) at the Massachusets Institute of Technology (MIT) presents an approach that combines drawing and concepts of traditional music notation and harmony. It allows the creation and modification of motifs easily by drawing lines and strokes on a sketch window. Each motif is mapped to a colour. Bowed and plucked modes can be selected for changing the quality of the timbre. A harmony line that runs through the center of each sketch window can control harmonic progressions in major/minor and fourths harmony types.

The authors described the system thus:

> “Hyperscore, a Windows application written in C++ using DirectX, consists of an expansive, zoomable canvas where users can create any number of musical fragments and whole pieces. Users can position these musical objects anywhere on the canvas and can view four different levels of zoom for ease of editing. The first step in composing a piece is creating some melodic material in motif windows. The window’s vertical axis represents pitch (spanning two octaves), while the horizontal axis represents time. Users can stretch or shorten the window depending on how long the motive is. Colorful droplets represent notes, and users add them by clicking on the grid. The system interprets blank spaces as rests.
>
> The user chooses a color for each motive and composes a piece by selecting different colored pens and drawing into a sketch window (see Figure 17a). Every time the user draws a line of a particular color, Hyperscore adds the motif mapped to that color to the piece. The start and end points of the line determine how many times a motif repeats. That is, a fixed pixel-to-duration metric calculates the length of time a line plays. Drawing the line straight makes the motive repeat with the precise melodic intervals of the original motivic material. Curves and bends in the line impose a pitch envelope on the motif’s repetitions but do not alter the melodic contour to the point that the new material is unrecognizable from the original motif. Users can reshape the lines after drawing them by right clicking and then dragging…All sound output is MIDI, and either the computer’s sound card or an external MIDI synthesizer can act as the output device”.
>
> (Farbood et al., 2004)

The method in which *Hyperscore* addresses harmony is described thus:

> “In the simplest example, harmony can be a single chord without a reference point, without regard to what precedes or follows it. Users can add individual chords consisting of three simultaneous voices to the sketch window. They are displayed as colored droplets, with each color representing a different harmony type: major, minor, augmented, diminished, and so forth.
> Defining transitions from one chord to another is the first step toward adding

> functional harmony. This can be as insignificant as the prolongation of a previous chord or harmonic function or as far-reaching as a move to a new key. Hyperscore lets users describe these types of harmonic progressions by shaping a central line. Depending on whether the curves in the line are going up or down and depending on their shape, the computer chooses relevant chords".
>
> (Farbood et al., 2004)

Three harmony styles are implemented that are displayed in different colours:

- *"Diatonic"*: means that the computer changes all the chromatic pitches into "white" notes.
- *"Fourths"* harmony is based on chords built on perfect fourths as opposed to thirds.
- *"Major/minor":* indicates regular tonal harmony and can be most effectively shaped by the harmony line.

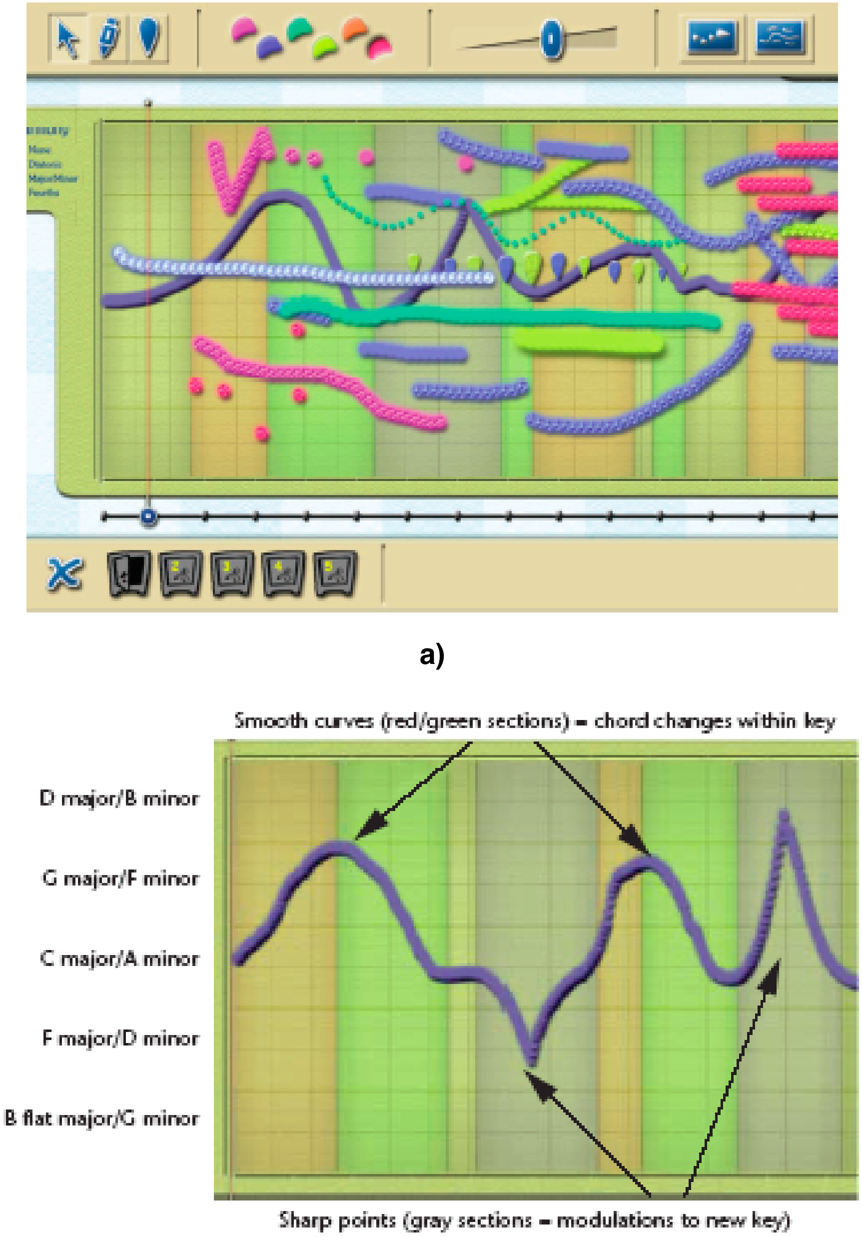
Changing the shape of the harmony line in this way controls the harmony (see Figure 17b):

- *Flat* areas result in stable chords (or tonic) areas in the current key.
- *Upward* areas result in unstable chords (or dominant) that require resolutions.
- *Downward* areas, which naturally follow upward areas, resolve the previous unstable harmonies.

Strengths

- Straightforward drawing metaphor.
- Control of the harmony by selecting different harmony types and changing a harmony line.
- Composition capabilities: events organization (rhythmic and harmonic grids, global timeline) and modification.

- Bowed and plucked modes for the quality of the timbre.



a)

b)

Figure 17: a) Hyperscore screenshot showing four motif windows and one sketch window b) Harmony line (from Farbood et al., 2004).

Weaknesses

- It doesn't support improvisatory performance of sound. There is no sonic feedback in real-time.
- It gives the user general (high-level) rather than detailed specification and control over the sound and graphics.
- The system lacks flexibility in mapping between image and sound.

- Just one kind of tool for drawing is supported.
- Most of the products of the interaction are determined by the system. The expressiveness derived from the gestural input is very limited.

Hyperscore allows the generation of melodies that sound “good” by creating motifs from free-hand drawings. In this way novice users can create basic compositions. However, these compositions are restricted to a set of predefined harmonic rules and simple visualizations that limit the expressiveness derived from the performer/composer gestures.

3.3 MetaSynth

MetaSynth is a spectrogram-based drawing system for the analysis and resynthesis of sound. The systems based on this technique, called *pattern playback,* take a digital image to represent the intensity of different audio frequencies over time, which is then used as a “score” for an additive or inverse-FFT synthesizer. The principal advantage of using this approach is the variety of sonic and visual results it can produce. Various systems similar to *MetaSynth* such as *RGS* (Real-time Graphical Synthesis) by Henry Lowengard for the Commodore Amiga (Lowengard, 1994) and *Coagula* by Rasmus Ekman for Windows machines (Coagula website, 2004) have been developed since the mid-1980’s.

The functioning of *MetaSynth* is explained as follows:

> “MetaSynth plays a picture and/or mark by scanning it from left to right. Every pixel within the image literally becomes an oscillator, or tape recorder, with control over volume, pitch, sustain, stereo placement, duration and the envelope of a sound. A time value is assigned to MetaSynth's horizontal axis, frequency, or pitch, is mapped to the vertical axis, brightness of pixels represents volume or amplitude, and pixel colour denotes the placement of the sound (pan position) within the stereo field…MetaSynth's Image Synth is reminiscent of a MIDI sequencer's piano roll display, in the sense that time is denoted by the horizontal axis and pitch by the vertical” (UI Software 1998).

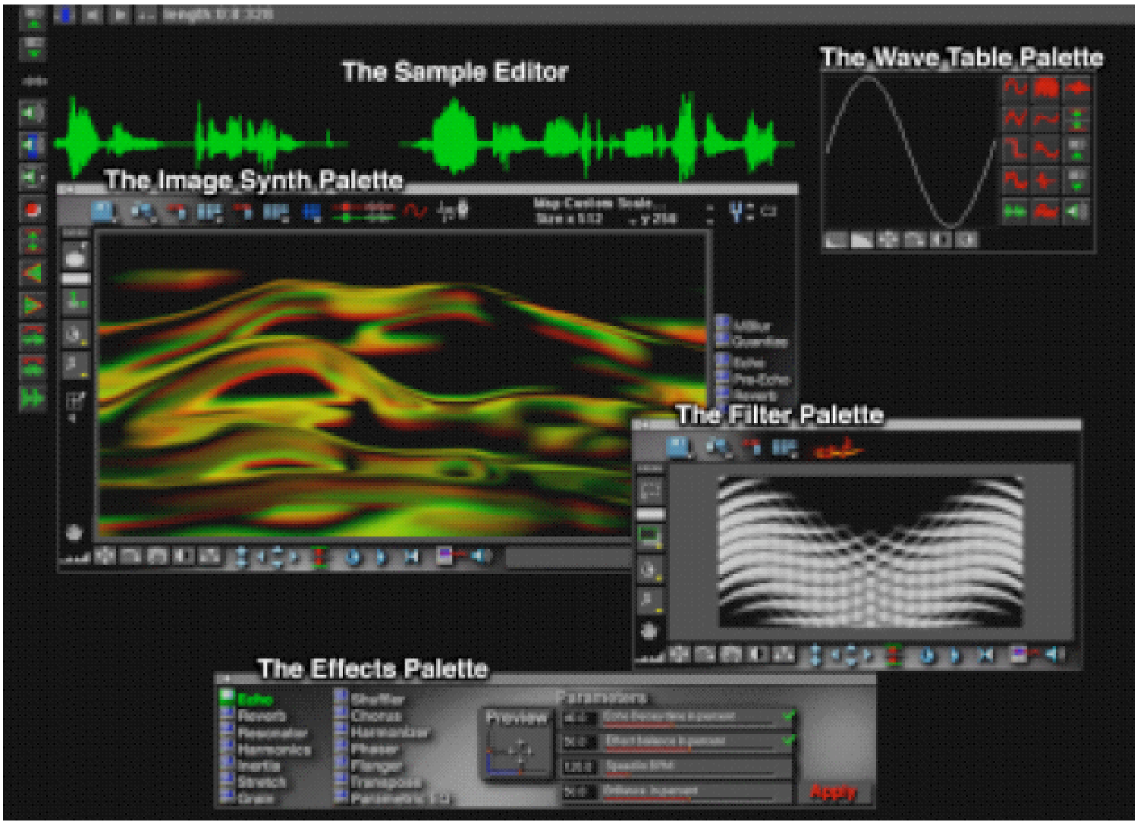


Figure 18: A screenshot of MetaSynth's palettes. Visual patterns (*Image, Filter, Wave Table*) can be created by using familiar paint tools (UI Software 1998).

Strengths

- Suitable for a meticulous style of sound design and composition.
- Different audio synthesis techniques available (Wave Table, FM).
- Filtering and real-time control and modulation of effects available (reverb, delay, flanging, resonators, granular synthesis among others). These affect both image and sound.
- A set of diverse and familiar drawing tools has been implemented.
- Rhythmic and harmonic grids have been implemented.
- It maintains the painting metaphor for the interaction across different palettes (*Image, Wave Table, Filter*).
- It possesses multitrack capabilities if linked with another software system called *Metatrack* (UI software).
- It can render results to audio files.

Weaknesses

- No real-time (improvisatory) performance capabilities are supported for the creation of sound. The sonic feedback is not immediate; the user has to playback the sound after he/ she paints into the spectrogram.
- It is not very intuitive. The step-by-step learning process is difficult.

3.4 Videodelic

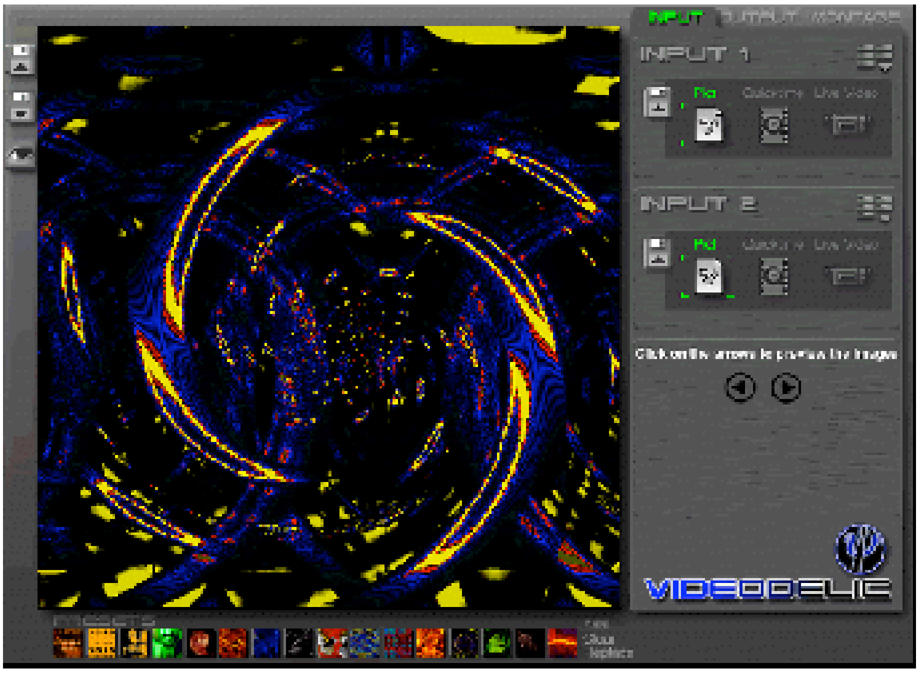


Figure 19: A screenshot of *Videodelic's* Input Window (UI Software, 2000).

Videodelic also developed by UI Software (2000) as a real-time video synthesizer optimized for live playing rather than as an audiovisual instrument for creating both sound and graphics from scratch. Controlling visual effects with an incoming sound links audio and graphics. The MIDI input provided in *Videodelic* also offers the possibility for connecting musical information to dynamic visuals; it permits the assignment of continuous and discrete MIDI controllers in order to modify diverse aspects of the graphics such as colour and effect parameters. The computer keyboard and mouse can also be used as controllers expanding the range of interaction alternatives.

Still images, movies or live video input are the source for the creation of real-time 2-D animations.

Strengths

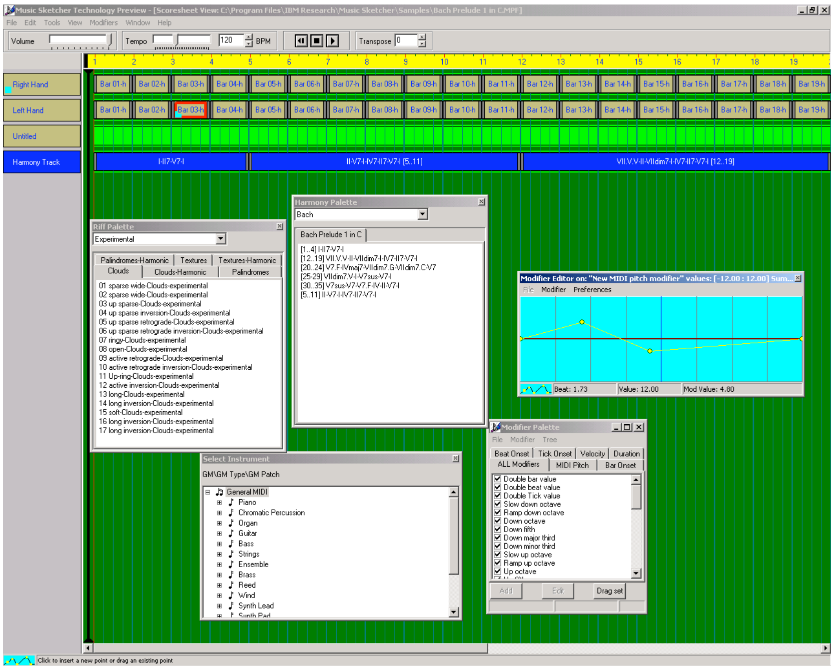
- Facility for creating structured visual compositions.
- Supports several kinds of visualization and effects.
- Several types of controllers for real-time parameter control are supported (audio input, mouse position, built-in LFO, MIDI input).
- MIDI control over global functions (change pictures, effects and display options).
- Generation of real-time 2-D animation from diverse sources (still images, movies and live video input).

Weaknesses

- Relatively small control over the visualizations presets. Most of the products of the interaction are determined by the system.
- Cannot generate images from scratch.
- It does not permit control over sound.

3.5 Music Sketcher

Music Sketcher is an environment for musical composition developed at the Computer Music Center at IBM Research (IBM Research, 1988). Like *Hyperscore* it presents a visual system that mediates between concepts of traditional notation and interactive visualisation.



Click to insert a new point or drag an existing point

Figure 20: A screen shot of Music Sketcher.

Music Sketcher follows a design philosophy that views all activities associated with computer music as a whole (Oppenheim, 1992). Similar to other systems such as *DMIX* (Oppenheim, 1992)*,* a major goal in the design of the system is to address and enable the composer to interact with his musical ideas while communicating with the full context of his music. *"This is achieved by simultaneously having different ways of visualizing and manipulating the music on every level: from the lowest level of the sonic event to the highest level of the composition's hierarchy"* (Oppenheim, 1992). *Music Sketcher* focuses on three technologies described by the system's designers thus:

- *Riff blocks*: small blocks of musical content
- *Modifiers*: real-time transformation algorithms
- *Smart Harmony*: Constrain pitches transformations within a chosen harmonic framework.

The general principles of the system's functioning are explained thus:

> "Music Sketcher allows you to compose music by dragging riff blocks and positioning them on a graphical metaphor for a score part within a score sheet. The music can be shaped by applying modifiers (graphical curves that alter various aspects of the music) to produce a wide range of musically expressive transformations on the pitch, articulation, and loudness of the music. Smart

> Harmony is a model of tonal music that constrains these riffs within the context of a selected chord progression“ (IBM Research, 1998).

At the time of evaluating this system only a few *Riff Blocks* were implemented without the possibility to modify them or create new ones. The actual potential of the system, therefore, remains as a subject of future experimentation.

Strengths

- Supports composition capabilities: event organization (rhythmic and harmonic grids) and modification.
- It is easy to learn the basic principles of operation.
- Real-time transformation algorithms for the audio.

Weaknesses

- Most of the products of the interaction are determined by the system.
- No real-time (improvisatory) performance capabilities for the creation of images and sound are supported.
- Detailed gestural specification and control over visual and sonic parameters is not possible. The expressiveness, therefore, is reduced.
- The system lacks flexibility in mapping between image and sound.

3.6 FMOL

Sergi Jordà's *FMOL* (Jordà, 2000) is a system that presents a closed feedback loop between the sound and the graphics, *the same GUI works both as the input for sound control and as an output that intuitively displays all the sound and music activity"* (Jordà, 2003).

FMOL is a playable instrument, rather than a compositional environment and for that reason the user cannot edit performances or trigger pre-recorded sequences while improvising. It is therefore hard to play a fixed sequence of pitches or a precise rhythm, as *"the interface is good for large-scale or statistical control but poorer for detailed specification"* (Jordà, 2003). However, the large number of sound synthesis algorithms (generators and processors) makes *FMOL* a very flexible system in terms of sound generation.

Jordà explains the process of interaction with the system thus:

> “In its rest position the screen looks like a simple 6x6 grid or lattice. Each of the six vertical lines is associated with one voice generator (*FMOL’s* sound engine supports six real-time synthesized stereo audio tracks or channels), while the horizontal lines are associated with the effects processors (filters, reverbs, delays, resonators, frequency, amplitude or ring modulators, etc.), embedded in each track. All of these lines work both as input devices (controllers) that can be picked and dragged with the mouse, and as output devices that give dynamic visual and sonic feedback” (Jordà, 2003).

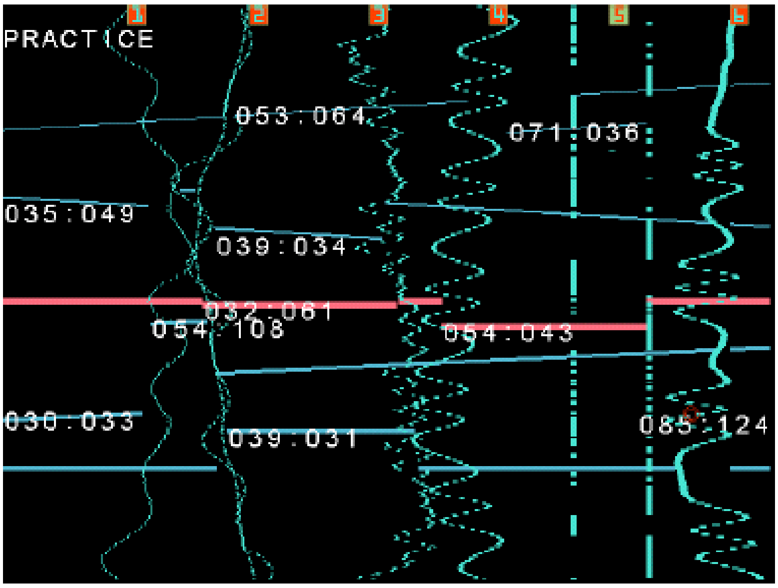


Figure 21: A screen shot of FMOL in action.

Strengths

- It allows rich and intricate control of sound parameters.
- Real-time (improvisatory) performance capabilities are supported for the creation of sound.
- Almost every feature of the sound synthesizer is reflected dynamically in the graphic interface.
- It has flexible mappings for image and sound through a modular design.
- Supports six real-time synthesized stereo audio tracks or channels.

- Several real-time processors can be applied to the sound.

Weaknesses

- It is not possible to edit performances.
- It does not allow detailed specification and control over the sound and image parameters.
- The graphics feedback parameters are predetermined by the system's designer and cannot be changed.

3.7 Summary

All the audiovisual systems reviewed here reveal the diversity of approaches and imaginative solutions researchers have developed to explore relationships between image and sound in the context of software-based systems for composition and performance. These systems range from approaches based on concepts of traditional notation and MIDI sounds, to systems that use abstract visual forms as input parameters for the generation of synthesized sound and/or music. An examination in Table 2 shows that none of the systems I have reviewed posses all the attributes that comprise my "ideal system".

System	**Real-time performance**		**Composition capabilities**		**Expressivity**		**Modifiers, FX & filters**		**Modul -arity**	**Learnability**	
	Sound	Image	Sound	Image	Sound	Image	Sound	Image		Easy	Difficult
Yellowtail	X	X			X	X		X		X	
Loom	X	X			X	X					X
Warbo	X	X			X						X
Aurora	X	X								X	
Floo	X	X				X					X
FMOL	X	X			X		X		X	X	
Hyperscore		X	X				X			X	
Music Sketcher			X				X			X	
MetaSynth		X	X	X		X	X	X	X		X
Videodelic		X		X				X	X	X	
IDEAL SYSTEM	X	X	X	X	X	X	X	X	X	X	

Table 2: Comparison of different AV systems.

However, some of them are a fairly close match to this “ideal system”, e.g. *Metasynth* (UI Software, 1998). The biggest drawback of *Metasynth*, from the perspective of this research, is the lack of real-time performance of sound, a property that is present in both Levin’s and Jordà’s systems, which also incorporates dynamic visual feedback. Other systems, which are much closer to, although more permissive, than sequencers or score based specification, are *Hyperscore* (Farbood and Pasztor, 2001) and *Music Sketcher* (IBM Research, 1998). These both make use of the *timelines and diagrams* metaphor but also use alternative ways of control and generation of audiovisual material such as drawing strokes that are mapped to structural elements in the music in *Hyperscore* or insertion of small blocks of musical content in *Music Sketcher*. However, the audiovisual outcome of these last two systems has a reduced expressiveness. Other painterly systems such as *Floo* and *Aurora* (Levin, 2000), while they allow the generation of interesting and variable images and sounds, are limited in that most aspects of the interaction are predetermined by the system.

Table 3 summarizes the different synthesis techniques used by the systems I have reviewed. The diversity of algorithms and the differences in the realisation of timbre reflect the possibilities of diverse approaches to timbre as a core element in achieving expressiveness and sonic variety.

Another core issue for audiovisual systems design is the way the user controls the method of synthesis. Table 3 compares the different systems and it shows that the mouse is the most common controller. Though the mouse is ubiquitous, it does not permit the capture of a wide range of movements executed by the user. Only *Loom, Warbo, FMOL* and *Videodelic* allow the use of different kind of controllers such as drawing tablets and MIDI controllers.

A *physical controller* with multi-parametric control and high-resolution capabilities is a desirable feature that an interactive AVS should provide. This is essential if the aim is to build a system that can take advantage of the nuances and variety of the user’s gestures and expand and augment them to achieve expressive results in an audiovisual environment. However, it is important to say that the separate design of controllers and generators may have drawbacks. Jordà has written several papers on this issue (Jordà, 2002a & 2002b).

The *simultaneous* generation of synthesised sound and image, which is characteristic in some painterly systems (e.g. *Loom*), is very useful in achieving expressive effects. This allows direct control of synthesis parameters. The use of a pointing device that follows the trajectories of the user's movements, in an expressive painterly system, makes sense if it also allows high-resolution and multi-parametric control (e.g. drawing tablet and pen).

Flexible mapping between image and sound implemented in some systems such as *FMOL* and *Metasynth* also expands the expressive range and the variety of the audiovisual composition. A global organisation of events along a timeline, which is implemented in some of the systems reviewed (e.g. *Music Sketcher, Hyperscore, Metasynth),* is also an important element in organising compositions.

In summary, the most important properties, that I have derived from the existing systems evaluated in this chapter, to build an expressive and flexible audiovisual system for performance and composition are:

- Gestural input.
- Simultaneous generation of sound and image.
- Multiple mappings between image and sound.
- Global organisation of events along a timeline.

System	**Image**		**Sound**		**Controllers**
	Static	Dynamic	Synthesis Technique	MIDI	
Yellowtail		X	Spectogram-based additive synthesis oscillators.		Mouse, keyboard
Loom		X	Frequency Modulation (FM)		Mouse, Wacom tablet
Warbo		X	Waveshaping (Chebyshev polynomials)		Mouse and Wacom tablet
Aurora		X	Granular		Mouse
Floo		X	Granular. Shepard tones.		Mouse, keyboard
FMOL		X	More than 100 synthesis algorithms		Mouse, keyboard, MIDI controllers
Hyperscore	X			X	Mouse, keyboard
Music Sketcher	X			X	Mouse, keyboard
MetaSynth	X		Spectogram-based FM and Wave table		Mouse, keyboard
Videodelic		X		X	Mouse, keyboard, audio input, any MIDI controller

Table 3: Controllers and techniques for generating image and sound.

Chapter 4. The design of Flexible and Expressive Audiovisual Systems

4.1 Introduction to approach

Most existing software environments that present some kind of correspondence between image and sound in real-time are not very interactive. Among them are various popular media players (e.g. *Itunes, WinAmp, Real Player,* etc.) that have many possible dynamic visualizations (visuals) related to the music. Other systems such as *Videodelic* (UI Software, 2000) and *PixelToy* (LairWare Software, 2003) that are also called VJ (Video Jockey) Tools (Jordà, 2003), offer a restricted level of control because most aspects of the interaction are predetermined by the system.

Another approach involves developing or using a closed feedback loop between the sound and graphics. In this approach control, interactivity and expression play a very important role, and as we have seen are found in a variety of forms in systems such as *FMOL* developed by Jordà (2000) and *Loom, Aurora, Floo, Warbo and Yellowtail* developed by Levin (2000). These have been described in chapter 3 of this thesis.

Even when a closed feedback loop exists between sound and graphics, the question remains as how best to specify the characteristics involved, and how to make these expressive. Gesture is one obvious and very important source of expressiveness in human performance. The temporal and spatial characteristics of the gestures captured from an input device can be used to control diverse synthesis parameters for an audiovisual system. Alternative control interfaces with a tight relationship between gesture, image and sound can be implemented following this approach.

The approach described here also seeks to exploit complementary aspects of interaction. It explores various aspects of audiovisual performance and composition but has concentrated most of effort on the design of a system for the simultaneous generation and control of sounds and images in real-time using existing motion sensing technology. Figure 23 shows an outline of the proposed system. This scheme is similar to the one suggested by Wanderley (2001), shown in Figure 22, for a *digital musical instrument* (DMI), that he describes as an “instrument that contains a separate gestural interface (or gestural controller unit) from a sound generation unit” and both units are related via mapping strategies (Wanderley, 2001). However, the main difference between the system proposed and the scheme used by Wanderley’s DMI is the inclusion of an image generation unit.

This kind of system relies on using and interrelating multiple human sensory modalities (multi-modal interaction). Here “use” implies the task of adequately interrelating or coordinating the outputs across modalities so as to facilitate or engender engagement involving different modalities. This is an important and interesting field of research considering the often limited input/output options available in current computer systems.

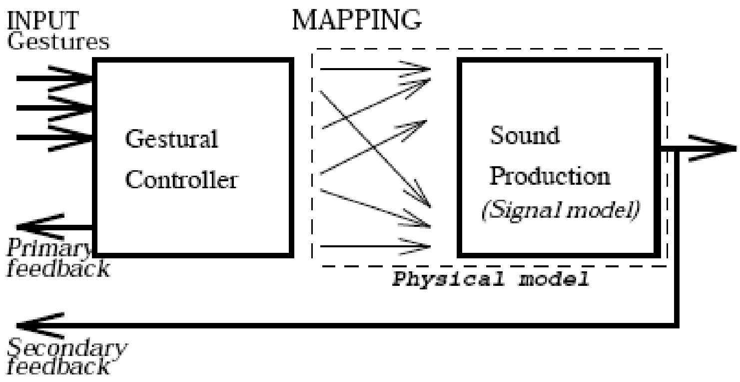


Figure 22: A Digital Musical Instrument representation (from Wanderley, 2001).

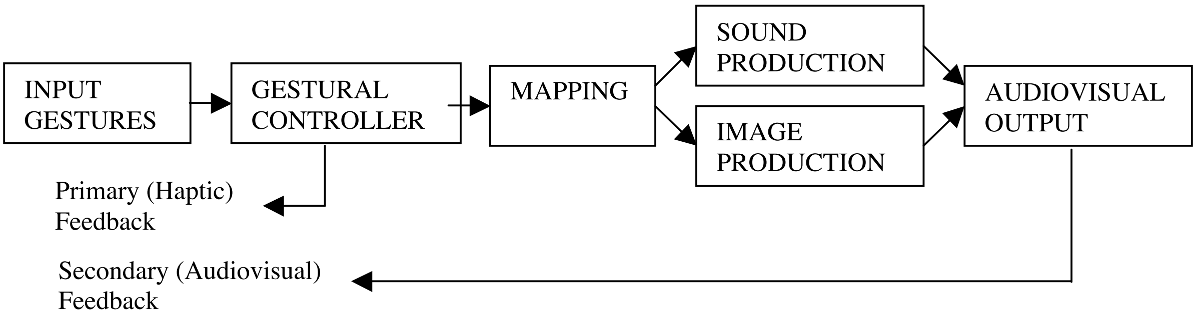


Figure 23: A flexible & expressive audiovisual system representation.

The main issues that I have isolated through evaluating various audiovisual environments and a review of related research for flexibility and expressiveness in the generation of software-based sounds and image are: *gestural control, dynamic visual feedback, sound synthesis, musical aspects and mapping flexibility.* I will discuss these issues and their implications for the design of expressive and flexible audiovisual systems using gestural input.

4.2 Gestural Control

One of the most interesting aspects of the development of software-based interactive environments, for musical or audiovisual expression, is the potential it offers for finding new ways of connecting various kinds of controllers and control strategies in correspondence to the user's or performer's needs. This open possibilities for performers with diverse physical and/or motor control abilities, gestural vocabulary, personal interests, social influences and cultural trends for experimenting with new ways of expression and/or to speed up the learning process in terms of the functioning of new instruments or systems. According to Mulder (1998):

> …there is a need for musical instruments with gestural interfaces that can adapt by themselves, through "learning" capabilities, or be adapted by the performer, without specific technical expertise, to the gestures and movements of the performer (Mulder, 1998).

Cook (2000) developed some principles for designing and constructing controllers for musical performance. These *"relate to practical issues for the modern instrument craftsperson/hacker"* (Cook, 2000). Some relate to human factors, others are technical. Quoting the author:

> "Some Human/Artistic Principles
> 1) Programmability is a curse.
> 2) Smart instruments are often not smart.
> 3) Copying an instrument is dumb, leveraging expert technique is smart.
> 4) Some players have spare bandwidth, some do not.
> 5) Make a piece, not an instrument or controller.
> 6) Instant music, subtlety later.
>
> Some Technological Principles
> 7) MIDI = Miracle, Industry Designed, (In)adequate.
> 8) Batteries, Die (a command, not an observation).
> 9) Wires are not that bad (compared to wireless).
>
> Some Other Principles
> 10) New algorithms suggest new controllers.

11) New controllers suggest new algorithms.
12) Existing instruments suggest new controllers.
13) Everyday objects suggest amusing controllers."

(Cook, 2000)

Various other aspects have to be considered within the subject of gestural control of computer generated sound and images for audiovisual expression including gesture types, simultaneous control of multiple parameters, input devices or controllers, timing, rhythm, user training and interaction modalities amongst others.

4.2.1 Defining Gesture

The meaning of gesture may vary significantly depending on the context. However, the diverse nature of gesture in the human computer interaction, music and performing arts contexts present concepts that are useful for this research.

According to Wanderley (2001), within the musical context the term gesture may have various meanings such as:

- "A composer may use the term *musical gesture* to designate a sequence of events within a space of musical parameters; sometimes it can also have some relation to a form of thinking – a movement of thought.
- A performer, on the other hand, may consider *performance gestures* as the technique used to play an instrument, where it encompasses not only the actions that actually produce an excitation or a modification of an instrument's properties, but also the accompanying body movements and postures.
- Computer musicians or computer music performers using electronic means to produce sounds may have a concept of gestures as isolated movements related to specific physical variables, such as pressure, velocity, acceleration, etc. that may be captured by sensors and transformed into digital signals input to computers."

(Wanderley, 2001).

More generally, performer actions can be divided into:

a) Actions where no physical contact with an instrument or device is involved and

b) actions where some kind of physical contact and/or manipulation of an object take place (Wanderley 2001, Mulder 1998).

The actions of the second type are those of principal interest for this thesis. In order to avoid confusions the term *instrumental gesture* (Wanderley 2001, Cadoz 1988) will be used as meaning instrument or object manipulation.

Cadoz (1988) proposes an instrumental gesture typology based on function that Wanderley (2001) summarizes as follows:

> - *"Excitation gestures*: these provide the energy that will eventually be present in the perceived phenomena. They can be either:
>   - *Instantaneous* (percussive or picking): the sound starts when the gesture finishes.
>   - *Continuous*: when both the gesture and the sound co-exist.
> - *Modification gestures*: are related to the modification of the instrument's properties, without any substantial expense of energy being transferred to the final sound. This modification affects the relation between the excitation gesture and the sound and therefore introduces another expressive dimension. It may be either:
>   - *parametric* (or continuous), when there's a continuous variation of a parameter, such as vibrato, for instance. It can be either continuous or discrete (for instance in a violin or a guitar, respectively.)
>   - *structural*, when the modification is related to categorical differences, such as the insertion/removal of an extra part of the instrument (a mute in the case of the trumpet, or a register in an organ).
> - *Selection gestures* are the ones that consist of a choice among multiple similar elements in an instrument. One can consider two possibilities: either sequential or parallel selection. This gesture differs from the previous ones (excitation and modification) in that it neither provides energy to the resulting sound nor modification of any of the instrument's properties."
>
> (Wanderley, 2001)

Within the context of multi-modal interaction in a virtual environment Choi (1998) proposes a classification of "*fundamental human movements that relate the human subject to dynamic responses in an environment*" (gestural primitives) in three types:

- *Trajectory-based primitives*: e.g. changes of orientation.
- *Force-based primitives*: e.g. gradient movements.
- *Pattern-based primitives*: e.g. quasi-periodic movements.

These various approaches to the definition of gesture give significant clues about the gesture characteristics that need to be taken into account for designing gesturally controlled audiovisual performance systems. These characteristics include *continuous excitation gestures* and *parametric modification gestures* related to *instrumental gestures* (Cadoz, 1998: Wanderley, 2001). The *continuous excitation gestures* are those present when both gesture and sound (and image in my proposed system) are simultaneous. In terms of real-time performance, this is very important for having detailed control of visual and sonic parameters. The *parametric modification gestures* described above allow making modulations on the sound (and image) adding expressiveness to the perceived results. The *instrumental gestures* and their distinctive multidimensionality and haptic feedback are amongst the most important characteristics for achieving expressive results and direct, detailed control in audiovisual systems.

4.2.2 Interaction Modalities

Similar to the definition of gesture, the term *interaction* has different meanings depending on its context. Wanderley (2001) considers the connotations related to computer music in which the term interaction may mean:

- *"instrument manipulation* (performer-instrument interaction) in the context of real-time sound synthesis control.

- *device manipulation in the context of score-level control*, e.g. a conductor's baton used for indicating the rhythm to a previously defined computer generated sequence.

- *device manipulation in the context of post-production activities*, for instance in the case of gestural control of digital audio effects or sound spatialisation.

- *interaction in the context of interactive multimedia installations* (where one person or many people's actions are sensed in order to provide input values for an audio/visual/haptic system)"

(Wanderley, 2001).

In the envisaged system some of the interaction modalities mentioned above can be separately used or in combination. *Instrument manipulation* is one of them because I am using gestural input through a physical controller. In this way I have direct control over real-time synthesis. I am also aiming for a score-level control, but not necessarily only through a device manipulation. This can also be achieved through other control strategies such as modification of diagrams, control panels and timelines on the computer's screen.

Wanderley (2001) also proposes the analysis of the human-computer interaction in music according to a set of basic features. Many of these are important in the design and functioning of a flexible and expressive audiovisual performance system:

- The connotation of interaction as *instrument manipulation* (performer-instrument interaction).
- Adaptability to different levels of expertise.
- The role of the computer system as a tool of expression.
- The use of contact actions.
- The expressive (artistic) role of the sound output.
- Both auditory and visual as primary communication channels.
- Hand gestures and manipulation.
- Various degrees of freedom.
- The control of continuous and discrete variables.
- Auditory, visual and haptic feedback.

I will develop these aspects in chapter 5, *Design and Implementation of Miró,*

4.2.3 Controllers

Mulder (1998) uses the concept of *control surface* to describe the means used to capture a user's movements in the context of *virtual musical instruments.*

> "In more practical terms, the control surface consists of sensors for human movement capture, actuators for tactile as well as force feedback yielding a

> haptic representation and last but not least a visual representation. The control surface outputs data that represents the movements and gestures. These data are turned into sound variations after processing. The control surface may change shape, position or orientation as a result of the application of these forces" (Mulder, 1998).

Following Mulder (1998) the important focus of such research is to improve the compatibility between performer and instrument through focusing on one or both of the following topics:

- *"Gestural Range* : Some research aims to expand the gestural range of existing instruments, to exploit unconventional gestures or movements or unused aspects of conventional gestures, so that the range of adaptation can be expanded, despite the fact that it would still be limited due to physical laws.
- *Adaptability* : Some research aims to find specific methods to make the musical instrument as easily adaptable (performer implements change) or adaptive (instrument implements change) as possible" (Mulder, 1998).

This area of research has resulted in the development of many kinds of *alternate controllers*. Many of them have been presented in international conferences such as *New Interfaces for Musical Expression-NIME* held every year since 2001*.* Mulder (1998) proposed a taxonomy of existing *alternate controllers* designs as follows:

- *"Touch controllers* (e.g. *Axio, Wacom tablets*): Most alternate controllers that expand the gestural range still require the performer to touch a physical control surface, usually fixed in space but sometimes carried around. Although any of these controllers can be adapted to meet the specific gestural needs or preferences of an individual performer, such adaptation is limited by the particular physical construction of the controller. An important advantage of touch controllers is their ability to provide a haptic representation.
- *Expanded range controllers* (e.g. *The "Hands", Lightning, Theremin*): These controllers may require physical contact in only a limited form, or may not require physical contact but have a limited range of effective gestures. Despite their expanded gestural range compared to touch controllers, the performer can always "escape" the control surface and make movements without musical consequence. The haptic representation of these controllers is reduced or even absent due to less physical contact.
- *Immersive controllers*: The alternate controllers with few or no restrictions to the movements are most suitable for adaptation to the specific gestural capabilities and needs of a performer. They often rely on the use of a Dataglove or Datasuit

> to track (nearly) all human movements of interest so that the feeling of immersion is created. For immersive controllers, touch feedback and/or force feedback can only be provided in very limited form, if at all, with current technology.
>
> - *Internal* (e.g. *The Biomuse*): Controllers with a control surface the visualization of which is the physical shape of the human body itself. Limb features like joint angles are mapped in a one-to-one fashion to sound or music parameters.
> - *External*: Controllers with a control surface the visualization of which is so different from the physical shape of the human body that it can be visualized by the performer as separate from his or her own body, although the visualization may be impossible to implement as a physical shape. Limb features may be complex (e.g. derived features like distance between two finger tips) and/or these features are mapped in a complex (e.g. non-linear or many-to-one) way to sound and/or music parameters.
> - *Symbolic* (e.g. *The Miburi*): Controllers with a control surface that is, due to its complexity, (almost) impossible to visualize or can only partially be visualized and which requires formalized gesture sets like sign language and forms of gesticulation such as used in conducting to operate. Gestural patterns are mapped to structural aspects of the music."
>
> (Mulder,1998)

Wanderley (2001) proposed a classification adding the controllers whose design follow an established instrument's one thus:

> - *"Instrument-like controllers*, where the input device design tends to reproduce each feature of an existing (acoustic) instrument in detail. Many examples can be cited, such as electronic keyboards, guitars, saxophones, marimbas, and so on.
> - *Augmented Instruments*, also called *Hybrid Controllers*, are instruments augmented by the addition of extra sensors. Commercial augmented instruments included the Yamaha Disklavier."
>
> (Wanderley, 2001).

Given the "large quantity" of controllers available in the computer music and performing arts context it makes sense to choose an existing design matching our requirements for flexibility and expressiveness rather than developing a new one from scratch. The characteristics that I am looking for are closer to those that characterize the *touch controllers* (Mulder, 1998). The envisaged controller should provide:

- Haptic representation.
- Adaptability to new gesture vocabularies.
- High level of resolution and accuracy.
- Real-time multi-parametric control.
- Continuous control or modulation.
- Discrete control might be useful for some tasks (e.g. sequence triggering).

4.3 Dynamic Visual Feedback

Some of the research and development of software-based systems that "*permit a user to gesturally create and perform, in one way or another, pure, animated abstract graphics*" (Levin, 2000), has been done by artists with some background as engineers or vice versa (e.g. Scott Snibbe, John Maeda, Toshio Iwai). As a consequence the visually aesthetic aspects of these systems play a very important role. This treatment of the graphics linked to the generation of sound is of particular interest for this research.

Antecedents to this approach can be found in the term *Colour Music* interpreted as the combined display of image and sound together (Levin, 2000). Various artists and music composers have experimented for many years with the correlations of colour/image and music/sound. A number of this examples are mentioned by Pressing (1997) within his discussion of the generation of non-audio displays from sound or music or musical performance information (*sonigenic* and *musicogenic* displays) :

> "Compositional antecedents for musicogenic display begin in the 19th century, with such things as the color coding of musical tones described by 19th century composer Nikolai Rimsky-Korsakoff, and the musically driven (but pre-scored) control of lighting and olfactory stimulation in the late works of composer Alexander Scriabin (1872—1915) (*Prometheus, Mysterium*). There is also a notated part for color in Arnold Schönberg's *Die Glückliche Hand*. Prerecorded multisensory experiences developed also in the cinema and were found in Morton Heling's *Sensorama* (1960). In more recent times, interactivity has become the norm, often implemented via MIDI "(Pressing, 1997).

Levin included an extensive review of visual/colour music experiments in his master thesis including treatment of performance instruments, abstract film, optical sound-synthesis (pre-computational era) as well as the prevalent schemas by which sound and image have been connected computationally (Levin, 2000).

Sonification is also likewise a useful concept; the functional inverse of *sonigenic* display and is often used for information transfer (Pressing, 1997). As expressed by Pressing (1997), *"in this procedure, data (normally without intrinsic audio significance) are mapped into sound, usually with the aim of better displaying multivariate relationships".* The *pattern playback* technique used in spectogram-based systems such as *Metasynth* (see section 3.3 of this thesis) exemplifies the use of this concept.

Additionally, new notation schemes are needed to represent the expanded palette and dimensionality of sound and the changing performance possibilities of computer music. Roads (1996) stated that these new notation schemes serve at least four functions. These functions are:

> a) "To aid the composer in visualizing a work during its creation – notation as an expressive medium.
>
> b) To specify parameters for sound synthesis – notation as synthesis specification.
>
> c) To convey instructions to a musician in concert – notation as performer guide.
>
> d) To serve as documentation – notation as a reading score for study and teaching purposes."
>
> (Roads, 1996)

In this research I have explored alternative representations that can give the player/composer indications of how sound synthesis parameters vary over time through dynamic visual feedback. For instance, if the thickness of animated marks or shapes is mapped to the intensity of the sound, the performer can both see and hear that thinner marks sound softer and become louder as they grow thicker. In this way the visualisation is not just a static representation of the control interface, but also an active, and engaging element of the composition, that of itself has a particular aesthetic value.

Within this approach the graphics or visualisations are not symbolic notations to be simply read by the users, but rather a representation and control input for the sound generated by synthesis algorithms. This reflects the active status of such representation in the actuation of the sound. In Common Music Notation (CMN) the gap between specification and performance has to be filled by the player. On the other hand, as expressed by Roads (1996) "*musicians who work with improvised music have little need of a strict notation system, except to document a recorded improvisation"* (Roads, 1996).

In terms of control and specification, dynamic image generation from *gestural markmaking* (drawing, painting) in software-based systems presents opportunities for adding expressiveness to the visual representation (e.g. Golan Levin's systems). It takes advantage of the implicit temporality of the gestures if a high-resolution physical interface or input device is used in order to capture the nuances of the user's movements.

The possibilities for expression by manipulating an electronic drawing device such as a *Wacom* tablet and stylus are endless. If we use this kind of device together with a drawing or drafting program, we can draw or paint whatever we want and can make as many diverse marks as we like possessing a great variety of characteristics. In this way visual expressiveness can be achieved by the "brushwork" (nature of the marks, shape, texture, sensitivity of the brush or input device) and the use of colour (brightness, intensity). These visual variations are the outcome from the signal analysis used to extract relevant information, from the raw temporal and spatial data, captured from the input device. Diverse models and simulations free from the traditional laws of physics can be shaped able to communicate different kinds of information.

Different strategies can be applied for situating the graphics in the two-dimensional space, as in the painterly schema proposed by Levin (2000), "*the visual material is not (necessarily) situated along a set of coordinate axes like in the score-based systems, but rather in the free-form visual structure of a dynamic abstraction*". One important question that arises from this kind of schema is: Can the visual output of the system be read as a painting or a score, or both? As I have stated above the ideal is that it should be both.

The visual feedback of a flexible and expressive audiovisual system should have the following characteristics:

- Dynamic or animated (temporal properties)
- Free form or abstract
- Gesturally controlled and generated
- Diversity of synthesis algorithms

4.4 Sound Synthesis and Musical Aspects

4.4.1 Real-time sound synthesis

To achieve expressiveness through sounds in a software-based system, it makes sense to use a method for controlling the generation of sound at a low-level. Without direct, detailed control the system is confined to a relatively crude mechanism such as MIDI. However, the aim of this project is to engender expressiveness. Sonic expressiveness can be achieved with real-time sound synthesis. In real-time synthesis sound can be shaped by variable control of the parameters of a synthesis algorithm. A plethora of sound synthesis methods (additive, subtractive, frequency modulation, formant synthesis, granular, waveshaping, etc.) are available due to the evolution of computer music and music technology among other areas.

In trying to engage the expressiveness of a human performer it is worth considering using a gestural controller (section 4.3.3) to control the synthesis parameters. If we link gestural variables captured from an input device manipulated within various degrees of freedom (e.g. relative/absolute position, pressure, velocity, tilt, etc.) to sound synthesis parameters (e.g. center frequency, bandwidth, amplitude, skew for formant synthesis) (Wanderley, 2001), we can control and create an effectively infinite range of timbres and manipulate their evolution over time. Different mapping strategies can be applied depending on the number of parallel control variables (multiplicity of control), the control modalities (discrete or continuous), the dimensionality of control and the physical variables carrying the information among other issues (Pressing, 1990).

4.4.2 Musical Aspects

As expressed by Cook (2002), *"music performance is structured real-time control of sound to achieve an aesthetic goal (hopefully for both the performers and audience). Thus, one obvious application area for real-time parametric sound synthesis is in creating new forms of musical expression".*

Many composers and musicians (e.g. Todd Winkler, Robert Rowe, Jean-Claude Risset, David Wessel, etc.) have experimented different approaches and techniques for creating music with interactive systems in the computer (e.g. generative algorithms, sequenced techniques, transformative methods, etc).

The main interest of this thesis, follows these previous approaches in looking for ways to take advantage of the computer's capacity for synthesizing sound in real-time under human gestural control; to add the expressiveness and temporality inherent in a performer's gesture, to create new sounds and organisational structures.

In particular it is focussed on the composition process based on the control of audiovisual sequences gesturally created through interacting with the system. In this way different methods of organization can be implemented in order to allow the user to create compositions based on his/her own decisions (online and offline), rather than structures determined by the system.

In order to build a musical structure it is necessary to organize the sonic events over time. If the musical approach is timbre-based, one might think of the generation of sequences as composed of series of micro-events along the evolution of a sound over time. This way expressive variation (timbral change) could be used to generate a structural timing (rhythm). For instance, if we generate and record a sound (sequence of micro-events) of any duration in an improvisatory way without a specific time reference (e.g. tempo speed), we can produce rhythms by re-iterating the recorded sound in different forms such as playing it back in a loop according to a time interval, perhaps that of its own period. Moreover, variations in duration created by shortening or lengthening it (expressive timing) or changing start and end points can create interesting textures that can be combined with silence (rests) and thereby generate a large variety of structures.

If we add more sounds (sequences, parts) in parallel, generated in a similar way and synchronized to a common and periodic triggering time, more sonic textures and rhythms can be created. For this, a global view of the sequences representation along a timeline will be useful for both understanding and controlling the temporal relations between elements. Other characteristics such as the relative amplitude (loudness) of sequences can also be modified (dynamic variation) in order to add expressiveness and control. The compositional process, therefore, exists at many different levels: from the generation (performance) of sounds (low-level) to the organization and synchronization of sequences over time (high-level).

A more traditional method is also possible. For example, if we have a time reference such as tempo speed, we can then start to generate sounds according to that tempo. This is similar to the performance of improvised traditional or popular music (e.g. rock setup with drums, bass, guitar, keyboards) where some instruments set the tempo (bass, drums) and the others (guitar and keyboards) build motifs and/or phrases according to that tempo. In this way the performer can explore different possibilities of a sound generation system by means of gesturally generated timbral variations and modulations, and control of duration (e.g. staccato, legato) and accent (e.g. piano, forte) by manipulating an input device according to a time reference. However, this method may need a long time for the performer to achieve a precise control of the system.

In order to produce more controllable results and facilitate the learning process, the designer of the system can predefine structures at different levels.

The exploration of particular sounds (processing and manipulation) combined with traditional music organisational notions such as rhythm, suggests new methods for expression and structure in software-based music.

Following the ideas discussed in this section, an expressive and flexible system for generating sounds should provide:

- Real-time sound synthesis capabilities.
- Control over dynamic variations.
- Gestural multi-parametric control.
- Different methods for playing back and modify recorded sequences.
- Events or sequences synchronization that permit the generation of rhythms.

4.5 Mapping Flexibility

Because there is no "objective" mapping from sounds to image or vice versa, flexibility in the mapping between the aural and visual dimensions is important if users are to search for mappings that they feel comfortable with for the audiovisual feedback from the gestural input. Wanderley (2001) states that by changing the mapping used "*the same gesture (cause) can lead to the production of completely different sounds or images"*.

Designing a system that contains a variety of graphics generation algorithms, which the user can explore, that match specific sound generators and vice versa, seems an appropriate strategy to realise the flexibility of mapping required, because it expands the expressive range and control of the system. In this way the performer/composer can decide what is associated to what in a personal, perceptually motivated way.

In achieving flexibility Wanderley (2001) found that the modularity of a system helped in organizing the accommodation of modes of control to different users needs. Quoting the author:

> "The modular concept of digital musical instruments (in a system called *Escher*) proves useful in choosing the interaction level desired by the user, i.e. depending on the user's technical skills or specific musical aims, one can arrange the mapping in order to allow different degrees of control complexity. It is strongly dependent on the controller one uses, e.g. number of available output parameters, nature of the available parameters - continuous/discrete, range, etc." (Wanderley, 2001).

Following the experience of Wanderley (2001), the implementation of a mapping switching mechanism controlled by the user based on his/her preferences is a desirable feature for an interactive system. Through modularity the mapping can be made entirely programmable and only limited by the synthesis models and their input parameters. Nevertheless the approach also lends itself to specifying of presets or defaults that could be programmed, in order to introduce and guide the novice user through the possible metaphors of interaction.

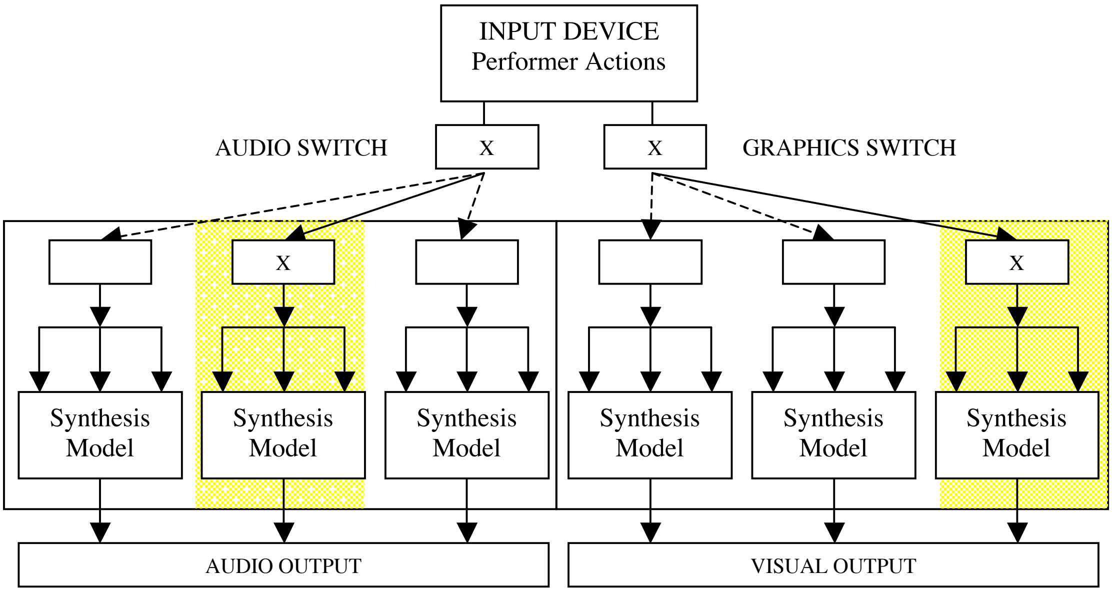


Figure 24: A representation of the mapping distribution. The coloured models are active.

The definition or distribution of the mapping across various levels, such as controller parameters, synthesis parameters and synthesis models can be derived from this approach. This is shown in Figure 24.

Using this mechanism the user can choose what synthesis models to apply and what controller parameters match to any synthesis parameter. Different mapping strategies can be applied to implement these modes that follows the classification proposed by Rovan et al. (1997) thus:

- *"One-to-One Mapping* : Each independent gestural output is assigned to one musical parameter, usually via a MIDI control message. This is the simplest mapping scheme, and consequently is often the least expressive.
- *Divergent (One-to-many) Mapping* : One gestural output is used to control more than one simultaneous musical parameter. Although it may initially provide a macro-level or general expressivity control, this approach is limited when applied alone, as it does not allow access to internal (micro) features of the sound object.
- *Convergent (Many-to-one) Mapping* : In this case many gestures are coupled to produce one musical parameter. This scheme requires previous experience with the system in order to achieve effective control. Although harder to master, it proves far more expressive than the simpler unity mapping."

(Rovan et al. 1997).

These strategies can be applied separately or in various combinations, as shown in Figure 25, depending on the expressive control aims and the user's skills. Also, as proposed by Mulder (1997) in the context of mapping virtual objects to sound parameters, suitable metaphors can be chosen in advance (presets) in order to "reduce the cognitive load, or in other words *is easy to understand*, for a novice user of the system" (Mulder, 1997).

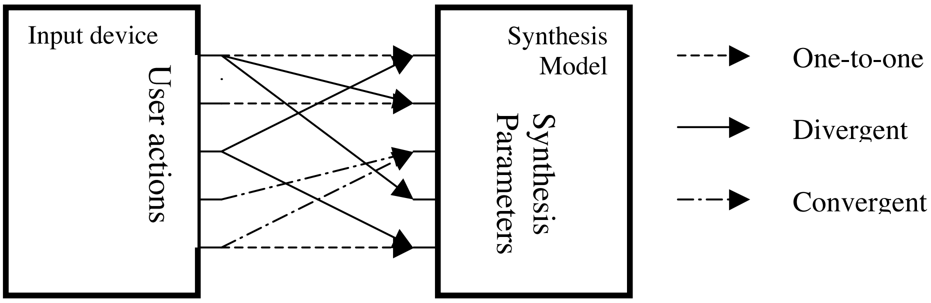


Figure 25: A representation of the mapping strategies.

4.6 Summary

In this chapter I have outlined a number of important issues for the design of expressive and flexible audiovisual systems using gestural input. Among them the use of real-time synthesis for both sound and graphics, presents interesting possibilities for achieving expressive audiovisual results and flexibility of control. Also, the use of dynamic images created by using gestural input for sound synthesis specification also offers attractive alternatives of interaction and expression.

Many gestural controllers have been developed in the context of computer music and performing arts that are capable of controlling interactive audiovisual systems. I am interested in those that offer haptic representation and multi-parametric control with high resolution.

Flexibility in an audiovisual system is to be achieved by having multiple sound and graphics generators connected through a programmable mapping. In this way the user can decide the mapping strategies to be applied in the system in order to adapt them to his/her own skills and needs.

The implementation of a sequencer the user can control for recording and playing back audiovisual sequences generated from a gestural input, is useful for creating compositions through different methods of synchronization and modification of sonic and visual aspects.

Chapter 5. Design and Implementation of *Miró*

This chapter describes the development of a gesturally controlled system, called *Miró,* (after the surrealistic painter Joan Miró) for performing and composing synthesised audiovisual pieces. The development of *Miró* has been the core aim of this research.

The system is suitable for the creation of abstract animations that can have similarities to the paintings of abstract painting movements such as the abstract surrealism (see section 2.1.4). Both visual and sonic outputs are primary communication channels for control and expression. In order to capture the user's gestures a *Wacom* graphics tablet and stylus have been used.

The system provides various levels of control, from the generation of "raw" audiovisual "data" controlled by the user's gestures, to a more detailed process of composition by using diagrams and control panels. Implementing a flexible mapping between *gesture, image, and sound* is one of the main goals of the prototype. Another very important issue explored through this prototype is the use of dynamic images to control and represent sound.

Miró implements a division of the interaction between the user and the system into two schemas: *performance schema* and *composition schema*. In the first schema the performer manipulates a *Wacom* stylus/tablet in order to generate audiovisual sequences. The flow of data in this schema is outlined in Figure 26. In the second schema the user organizes and plays the stored sequences using different methods through control panels, timelines and diagrams.

5.1 *Miró* – System Structure

Miró is implemented in Miller Puckette's *Pure Data* (PD) (PD, 2004), and the *Graphics Environment for Multimedia* (GEM) (GEM, 2004), a 3D graphics rendering package based on *OpenGL*, written by Mark Danks. These object-oriented visual programming environments allow the control and generation of audio and graphics in real-time.

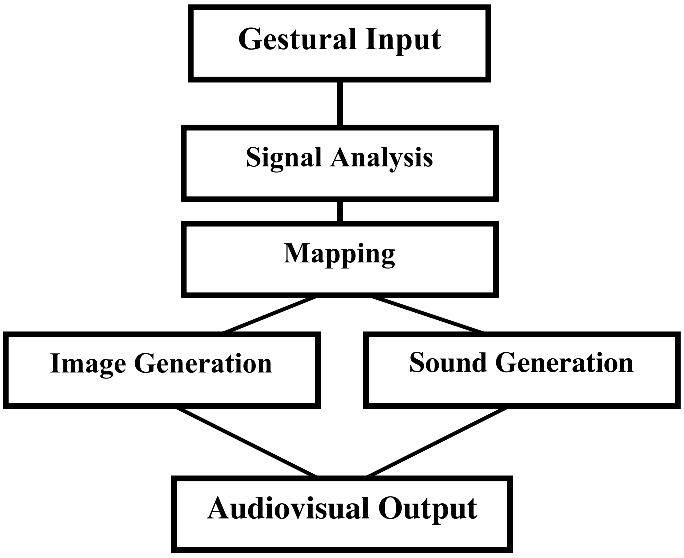


Figure 26: Outline of the performance schema in *Miró*.

The design of the system incorporates ideas found in previous environments such as *Loom* (section 3.1.2) for the visual and control concepts and *FMOL* (section 3.6) for the flexibility of sound generation in real-time. *Miró* uses a visual representation similar to *Loom* in which every visual element is associated with a corresponding sound-event and a timeline is *"wrapped around"* through the user's marks, this is linked to various graphics and sound generators through a mapping switching mechanism.

The system implements three different sound generators (two variations of *Frequency Modulation Synthesis* and a *Phase Aligned Formant* model) and three different graphics' generators or tools (*Paintbrush, Spray* and *Fountain*) that the user can assign to any of the four *Miro's* tracks at will. The outcome is, therefore, a multiple dynamic visual representation of different synthesized sounds organized in four parallel tracks.

In the prototype described here the gestural input is taken from a *Wacom* graphics tablet and stylus that allow the modification of the visual or sketch window by playing/drawing directly on it. The continuous representation of the gestures is both recorded and used to derive aspects of the user's marks such as the local velocity, pressure, X and Y coordinates. This occurs on a point by point basis. A diagram describing the input-output structure of a *Miró* track is shown in Figure 27.

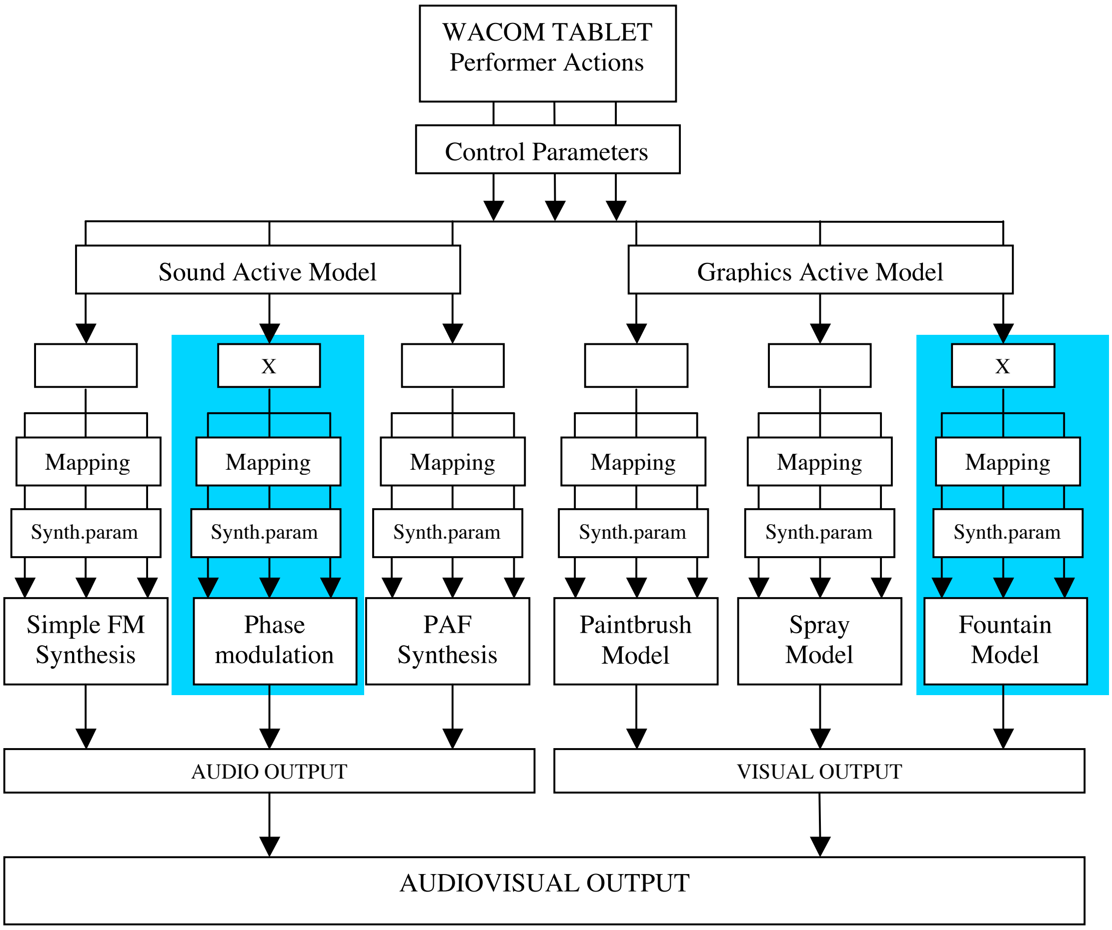


Figure 27: An input-output representation of a *Miró* track. The coloured models are active.

5.2 The Controller

As I have already stated, the design of a novel physical interface was not a goal of this research and therefore an existing controller has been used. I required an input device that is precise, accurate, available, and has various degrees of freedom.

Many of the available controllers for musical expression are based on the MIDI protocol (e.g. *The Hands*, *the aXiO*, keyboards, wind controllers, etc.) offering 7 bits of resolution and while their performance is acceptable in some setups, they have expressive limitations. These have been discussed by many authors (e.g. O'Modhrain, 2002; Moore, 1988). Given the requirements for the control of the visual and sonic representations and the capture of a wide range of gestures and nuances, the *Wacom* stylus/tablet controller seemed to be the best available controller for this particular prototype. It offers high-resolution (9 bits) absolute position sensing, detection of pressure and tilt, and data rates on the order of 100 Hz. Also, two buttons are available in the stylus for discrete control. Therefore, this controller is also suitable for the multi-parametric mapping requirements of the system. However, further experimentation with novel and alternate controllers will be desirable as part of future developments.

Many developers, composers and performers of real-time audio systems have used digitizing tablets as the input device (controller) for different designs, compositions and performance setups (e.g. David Wessel, 2002; Iannis Xenakis, 1986; Golan Levin, 2000; Daniel Arfib, 2002; Jacques Dudon, 2002, etc.). However, as we have seen, few of them have taken advantage of the potential visual representations that are inherent in the gestures (e.g. drawing, writing) that are intrinsic to using this kind of input device.

The use of the stylus/tablet as a drawing tool and sound controller together in some of Levin's audiovisual systems such as *Loom* and *Warbo* (sections 3.1.2 – 3.1.3), shows the advantage of having gesturallly created visual images linked to the control and generation of sound; *"the use of low level-level synthesis techniques permits the sound and image to be tightly linked, commensurately malleable, and deeply plastic"* (Levin, 2000). In this way the system's design broadens the communication and control channels with the user by exploiting visual and sonic feedback in different ways.

Having flexible mappings between control parameters and synthesis parameters allows experimentation with different relations between *gesture-image-sound* and the adaptation to meet the specific gestural needs or preferences of an individual performer.

5.3 Sound Aspects

Some methods of digital sound synthesis are more suitable for shaping sound in real-time and easier to manipulate than others. Among these, different methods of modulation synthesis and formant synthesis present interesting possibilities in terms of sound control flexibility and variety of expressive results.

Roads (1996) described modulation synthesis as follows:

> “Modulation” in electronic and computer music means that some aspect of one signal (the *carrier*) varies according to an aspect of a second signal (the *modulator*). To achieve the same complexity of spectrum, modulation synthesis is more efficient in terms of parameter data, memory requirements, and computation time than additive and subtractive synthesis.
> By changing parameter values over time, modulation techniques easily produce time-varying spectra. Carefully regulated modulations generate rich dynamic sounds that come close to natural instrumental tones. It is also possible to use modulations in a nonimitative way to venture into the domain of unclassified sounds.
>
> (Roads, 1996)

As expressed by Roads (1996), *“formant synthesis gives musicians a direct handle on one of the most important sound signatures: the spacing and amplitude of spectrum peaks”*.

According to Roads (1996), *“a formant is a peak of energy in a spectrum, which can include both harmonic and inharmonic partials as well as noise”*. Formant peaks are characteristic of the vowels spoken by the human voice and the tones radiated by many musical instruments.

Many techniques can generate formants including additive synthesis, subtractive synthesis, granular synthesis, frequency modulation, and physical modelling (Roads, 1996). However, some techniques such as FOF (Formant Wave-Function synthesis), VOSIM (originally used to model vowel sounds), WF (Window Function synthesis), and PAF (Phase Aligned Formant synthesis) were designed primarily for formant synthesis. Among these, we can find techniques aimed at real-time musical applications (e.g. PAF).

5.3.1 Sound Synthesis Techniques

Several of the methods mentioned above have been implemented in the *Miró* prototype. These synthesis models, whose parameters are controlled by the user's gestures, generate the sounds in *Miró* resulting in real-time synthesized stereo audio tracks.

I have explored the possibilities of PD's real-time sound synthesis capabilities by adapting three different sound synthesis algorithms allowing gestural control of parameters linked to graphic synthesis algorithms. These synthesis methods are: *Frequency Modulation (FM) using two cosine wave oscillators (simple FM), FM using a sawtooth generator and a cosine wave oscillator (phase modulation), and Phase Aligned Formant (PAF).*

5.3.1.1Simple Frequency Modulation (FM)

The diagram in Figure 28 and the patch in Figure 29 show the classical FM (*simple FM*) synthesis technique developed by John Chowning (1973). It comprises a cosine wave oscillator with vibrato controlled by another cosine wave "modulation" oscillator. To hear a sine wave with vibrato a carrier frequency is set, perhaps to 400 Hz or so, the modulation frequency between 5 Hz and 10 Hz and modulation index values between 0 Hz and 400 Hz. To get the FM sound, all three parameters carrier frequency, modulation frequency, and modulation index are set in the hundreds (Hz). A timbral change results as we sweep the modulation index, because this changes the amplitudes of the components of the output sound but not their frequencies. The component frequencies are equal to the carrier frequency, plus or minus multiples of the modulator frequency (PD documentation).

In Figure 28 the *modulator* and the *carrier* are both periodic and quasi-periodic oscillators with characteristic frequency, amplitude, and waveshape. In this case:

- The *frequency* of the *modulator* affects the *rate of change* of the *carrier's frequency*.
- The *amplitude* of the *modulator* affects the *degree* or *depth of change* of the *carrier's frequency*.
- The *shape* (or timbre) of the *modulator* affects the *shape of change* of the *carrier's frequency.*
- The *amplitude* of the *carrier* is not changed.

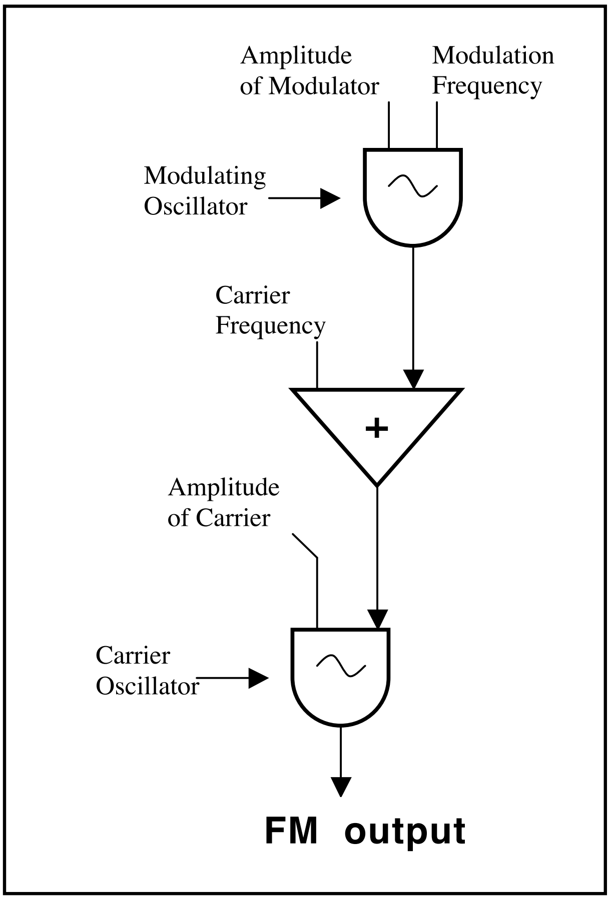
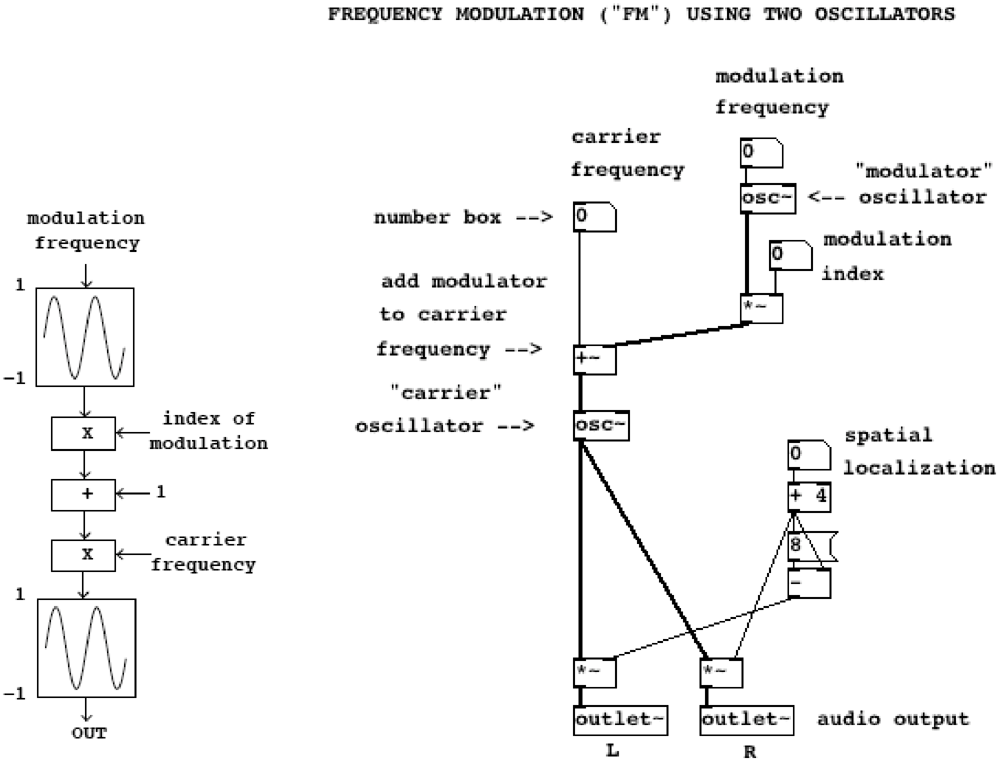


Figure 28: General diagram for simple FM synthesis.

In my implementation there is a spatial localization feature that by default takes the controller's values along its *x* axis to distribute the sound in left and right channels (L / R) accordingly. This makes visual and auditory sense but the user can arbitrarily assign any other control parameter to this feature. Therefore, this synthesis algorithm contains a binaural focus to convey information about the sound source location.



a) **b)**

Figure 29: a) Block diagram for simple FM synthesis (from Puckette, 2003) b) A screenshot of *Miro's* simple FM synthesis patch.

The “number boxes” in the patch shown in Figure 29 are the synthesis parameters that the user can assign to any control parameter from the *Wacom* tablet. These parameters are:

- Carrier frequency.
- Modulation frequency.
- Modulation index.
- Spatial localization linked to general amplitude.

Table 4 shows one of the mapping configurations that the user can use for this synthesis technique. By varying the stylus pressure slowly with this setup it is possible to get the effect of *tremolo* (slow amplitude variation) linked to timbral changes due to the parallel sweep of the modulation index. The carrier and modulation frequencies change at different rates by tracing trajectories with the stylus along the *Y*-axis generating higher frequencies as it reaches the top of the tablet and visual window. The displacement of the stylus from left to right and vice versa causes similar effects in the localization (panning) of the sound in the stereo field, creating two symmetric sonic halves (mirror) along the *X*-axis. The duration of the sounds depends on the duration of the gestures.

Gesture Measure	*Graphics Synthesis Parameter*	*FM Synthesis Parameter*
X (horizontal) position	Spatial Localization X axis	Spatial Localization L / R
Y (vertical) position	Spatial Localization Y axis	Carrrier Frequency & Modulation Frequency
Local Pressure	Thickness & Brightness	Modulation Index & Amplitude
Duration	Duration	Duration

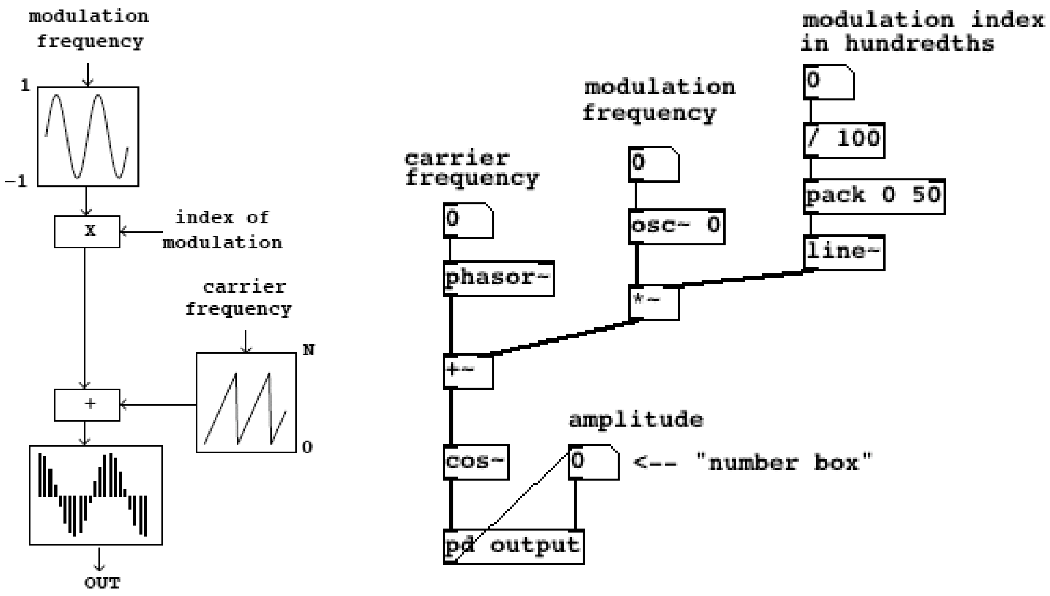
Table 4: *Miro's* FM synthesis mapping example.

5.3.1.2Phase Modulation

The patch shown in Figure 30 presents a variation of the *simple FM* technique. This variation consists in the substitution of the carrier frequency cosine wave oscillator for a sawtooth wave generator, and therefore the carrier frequency is a sawtooth signal. This technique is also known as *phase modulation*. To accomplish *phase modulation*, the carrier oscillator is split into its phase calculation (phasor ~) and its waveform (cos ~) lookup components. These together would be equivalent to a cosine wave oscillator (osc ~), but the "+ ~" between them adds the modulating oscillator's output to the phase. In this way the phase, instead of the frequency, of the carrier sinusoid is modulated sinusoidally (Puckette, 2003).

The modulation index, which in *simple FM* is in units of Hertz (Hz), is dimensionless for *phase modulation*. "Good" values tend to be between 0 and 1. In my patch the index of modulation is in hundredths. The "line ~" object is used to smooth changes in the modulation index (PD documentation).

Similar to the *simple FM* technique, to hear a vibrato the carrier frequency is set to 400 Hz or so, the modulation frequency between 1 Hz and 10 Hz and modulation index values between 0 Hz and 400 Hz. To get the FM sound, we set the carrier frequency and modulation frequency in the hundreds (Hz), and modulation index between 0 and 1 (PD documentation).



a) **b)**

Figure 30: a) Block diagram of FM synthesis realized as phase modulation (from Puckette, 2003) b) A screenshot of the FM synthesis (phase modulation) patch in PD with a sawtooth (phasor~) and a cosine (osc~) oscillator.

Table 5 shows one of the multiple mapping strategies the user can try. In this example the carrier frequency is constant and the modulation frequency changes by moving the stylus along the X-axis generating vibrato as it approaches the left side of the tablet and visual window. Moving the stylus along the Y-axis sweeping the modulation index produces timbral changes. The local pressure proportionally controls the amplitude.

Gesture Measure	*Paintbrush Tool Parameter*	*FM (phase modulation) Synthesis Parameter*
X (horizontal) position	Spatial Localization X axis	Modulation Frequency
Y (vertical) position	Spatial Localization Y axis	Modulation Index
Local Pressure	Thickness & Brightness	Amplitude
Duration	Duration	Duration

Table 5: *Miro's* FM synthesis mapping example.

5.3.1.3Phase Aligned Formant (PAF)

The PAF generator patented by IRCAM (l'Institut de Recherche et Coordination Musique/Acoustique) in 1993, *"is an inexpensive method for generating sounds with a desired pitch and set of formants; it is well adapted to real-time musical use because of its relative ease of computation, even in real-time situations where the formants and/or pitches required are not known in advance"* (Puckette, 1995). The PAF's timbral parameters are simply the center frequency and bandwidth; these can easily be changed over time with stable and predictable results. The PAF is a combination of a two-cosine carrier signal with a waveshaping pulse generator (Puckette, 2003).

The PAF block diagram shown in Figure 31 is separated into a phase generation step, a carrier, and a modulator. The phase generation step outputs a sawtooth signal at the fundamental frequency. The modulator is done by standard waveshaping. The carrier signal is a weighted sum of two cosines, whose frequencies are increased by multiplication (by k and $k + 1$, respectively) and wrapping. In this way the same sawtooth oscillator controls all the lookup phases. The quantities k, q and the wavetable index, are calculated as shown in Figure 32. They are functions of the specified fundamental frequency, the formant center frequency, and the bandwidth, which are the original parameters of the algorithm. The quantity p, is $1 – q$ (Puckette, 2003).

Puckette claims that compared to the output of the FM synthesis technique, the PAF's spectral evolution is much simpler. *"This greatly simplifies the problem of finding synthesis parameters to approach a desired sonic result"* (Puckette, 1995).

The patch shown in Figure 33 is the actual implementation of the PAF generator in *Miró.* The important controls are center frequency and bandwidth here controlled as MIDI values for better results following the PD documentation. The control inputs have been perceptually calibrated trying to achieve interesting and appealing sonic results. However, different mappings and calibrations are open to further experimentation.

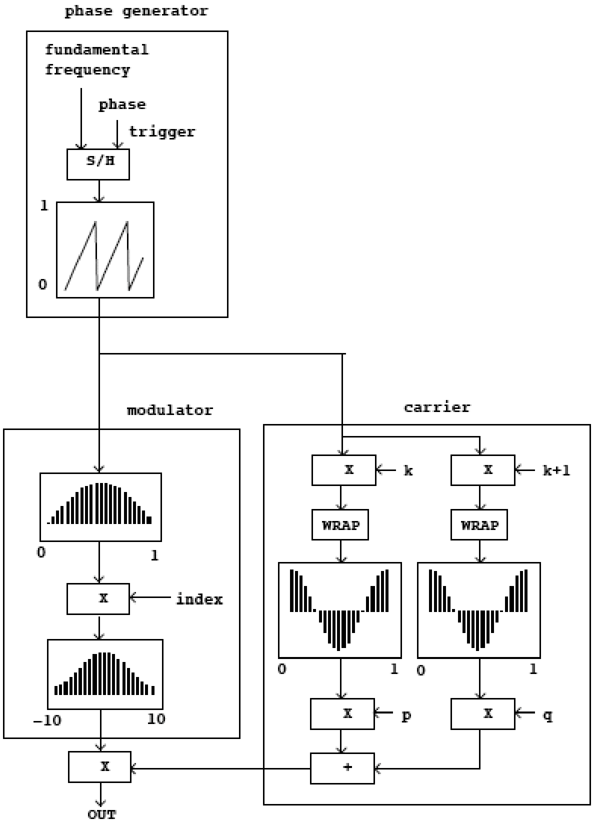


Figure 31: Block diagram of the PAF generator (from Puckette, 2003)

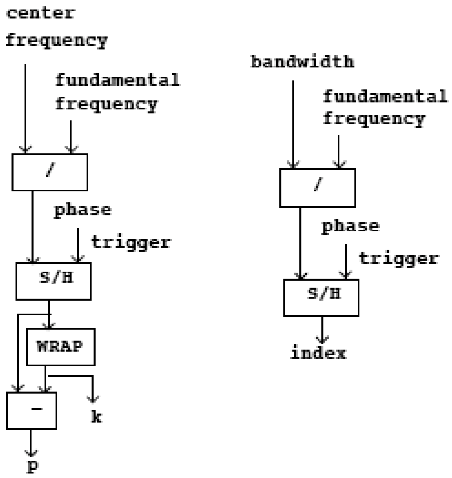


Figure 32: Calculation of the time-varying parameters, *a* (the waveshaping index), *k,* and *q* for use in the block diagram of Figure 31 (from Puckette, 2003).

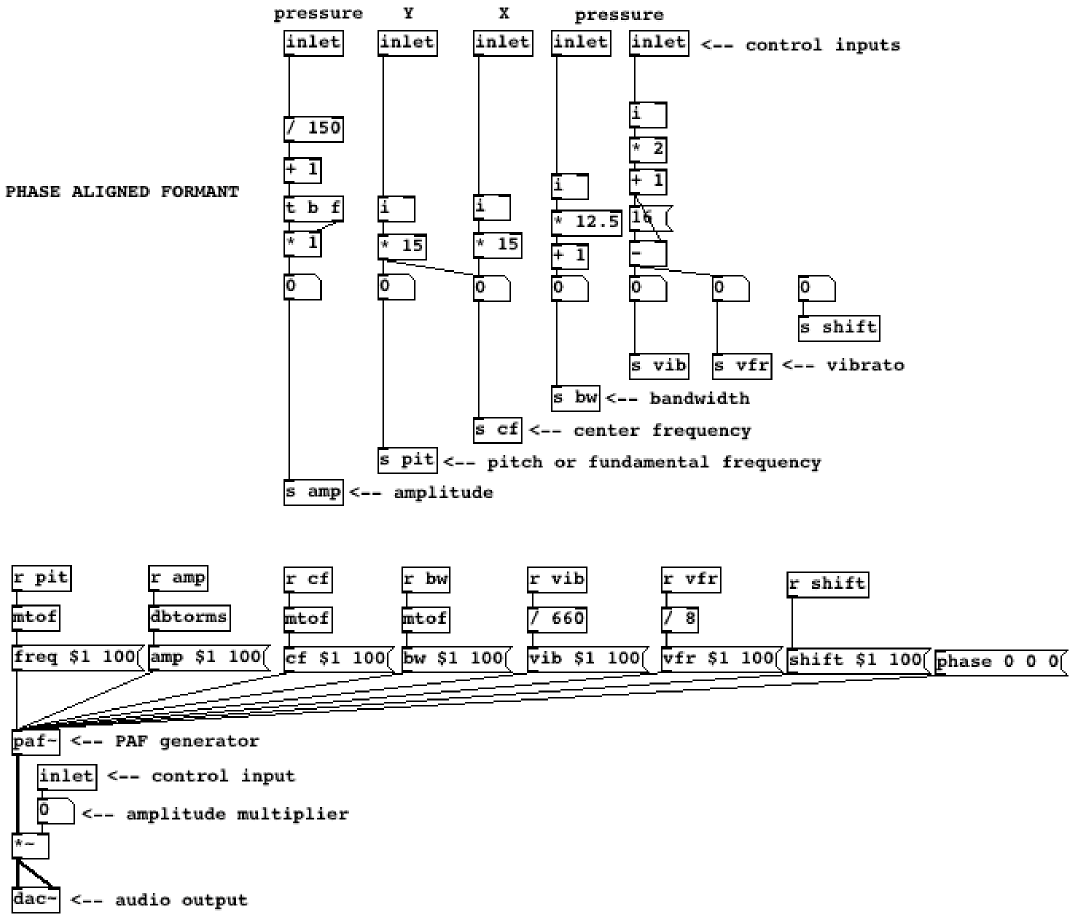


Figure 33: A screenshot of the PAF generator implementation in *Miró*.

The synthesis parameters that the user can assign to control parameters from the *Wacom* tablet are:

- Fundamental frequency
- Center frequency
- Bandwidth
- Amplitude
- Vibrato

Different mapping strategies are applied in the example shown in Table 6. Divergent mapping is present in this set-up between local pressure (control parameter) and three synthesis parameters bandwidth, vibrato and amplitude. This kind of strategy is also present between the stylus position along the *y*-axis (control parameter) and the fundamental and center frequencies of the synthesizer. A convergent mapping strategy is present between the stylus position in the *x* and *y* axes and the center frequency. This mapping permits the variation of various characteristics of the sound with a “single gesture” and therefore adds expressiveness to the sounds.

Gesture Measure	*Paintbrush Tool Parameter*	*PAF Synthesis Parameter*
X (horizontal) position	Spatial Localization X axis	Center Frequency
Y (vertical) position	Spatial Localization Y axis	Fundamental & Center Frequency
Local Pressure	Thickness & Brightness	Bandwidth, Vibrato & Amplitude
Duration	Duration	Duration

Table 6: A mapping example stylus-graphics-PAF (sound) synthesis.

5.4 Image Aspects

A dynamic visual shape generated by mark-making using an electronic drawing device may have any appearance. I am using abstract synthetic sounds and therefore the images or animations are abstract.

In this way the image is generated as a representation of the user’s gestures (pressure, trajectories, speed, angles of inclination, etc). The drawing tool metaphor (pen, brush, etc) used in popular paint programs such as *Photoshop* or *Paint* is the starting point for the generation of images in *Miró*. In this way the user can create a wide range of malleable visual forms.

The dynamic visual events properties to be controlled and modified are:

- Shape
- Colour Intensity (perceived brightness)
- Positioning
- Duration

5.4.1 Graphics Synthesis Techniques

The GEM (Graphics Environment for Multimedia) library, based on OpenGL, includes various objects that allow the creation of variety of visual results from the generation and variation of basic shapes such as squares or disks through to the complex control of particles behaviour reflecting physical models.. It is also possible to control the value for each colour component in the RGB (Red, Green and Blue) colour space. In *Miró* I am generating 24-bit colour images, in other words each RGB colour component has 256 values permitting a good level of resolution in order to achieve expressive images through continuous colour modifications.

Before proceeding it is necessary to introduce some functioning principles and components of GEM (more detailed information can be found in GEM's documentation) useful for understanding *Miro's* architecture:

- The GEM window shows all the rendered drawings.
- The *gemwin* object controls the window manager. It passes various messages to the manager, controlling the attributes of the window such as window-creation, background colour, enable/disable rendering and buffering, and number of frames per second to render at.
- The *gemhead* object connects the GEM objects to the window manager. The start of any *gemList* (different objects connected) begins with the *gemhead.* Without the *gemhead*, GEM objects will not receive the render command.
- The *gemtablet* object responds to events of a graphics tablet.

I have explored the possibilities of GEM's architecture by adapting three different graphics synthesis algorithms or tools (*paintbrush, spray and fountain*) allowing gestural control of parameters linked to the sound synthesis algorithms presented before. These graphics synthesis algorithms have a common characteristic, which is that *a timeline is "wrapped around" the user's marks,* similar to the design of Golan Levin's *Loom* (2000).

5.4.1.1The Paintbrush Tool

This is a simple algorithm, shown in Figure 34, that permits gestural control of three kinds of parameters:

- Colour parameters
- Position parameters and
- Shape parameters

The visual results are similar to those of using a paintbrush or a soft drawing tool. Diverse kind of marks can be created by having variations in colour brightness and mark (shape) thickness in different trajectories as a real-time representation of the user's gestures captured from the *Wacom* stylus/tablet. The GEM window shows these marks until the buffer is emptied by pressing the middle button of the stylus or by clicking the *restart* button in *Miró's* main control panel.

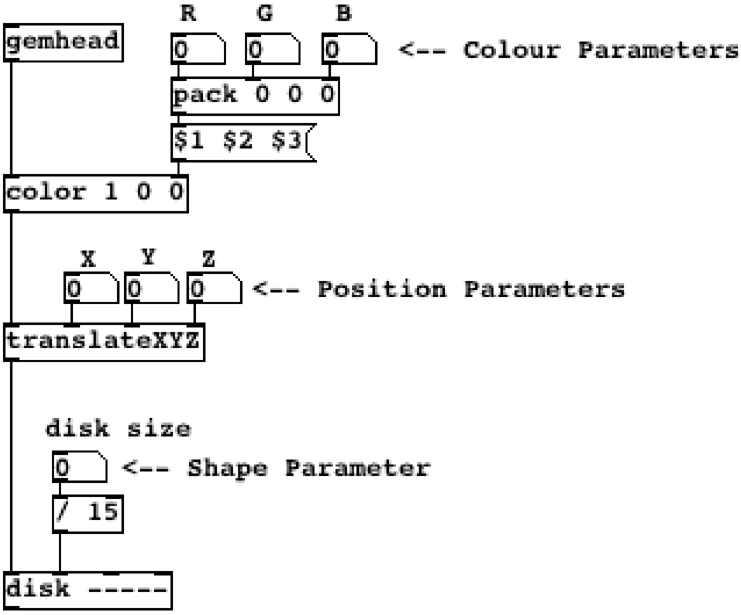


Figure 34: *Miró's Paintbrush* synthesis algorithm.

The graphics synthesis parameter section shown before in Table 6 corresponds to the paintbrush tool parameters. With this mapping, marks' thickness and brightness vary according to the pressure applied by the user, the trajectories represented in the screen are similar to the ones described by the user's movements within the tablet and the duration of the visual events are proportional to the gestures duration. An image captured from *Miró* in use, in Figure 35, shows the effects of these mappings.

Figure 35: A screenshot of diverse visual forms created with the *Paintbrush* tool.

5.4.1.2The Spray Tool

This algorithm, shown in Figure 37, makes use of some GEM's particle generation objects.

The *spray-like* appearance of the images created with this tool is due to the specification of various particle characteristics such as:

- Particle size.
- Velocity sphere: sets a sphere with a specified radius and midpoint to be the velocity-domain of newly emitted particles within the system.
- Particle source: the initial (variable) argument gives the number of particles that are emitted at each rendering-frame.
- Orbit point: makes the particles orbit about the position x, y, z. It has a gravity argument that determines how attracted the particles are to the point.
- Kill old: kill all particles that are older than the kill time. By incrementing the time the particles live longer and a shorter time removes them quicker. This makes the drawing more and less transient.

- Particle draw: finally draws a particle system set up with the other objects. There are line and point drawing modes. For the *spray* tool I am using the point mode.

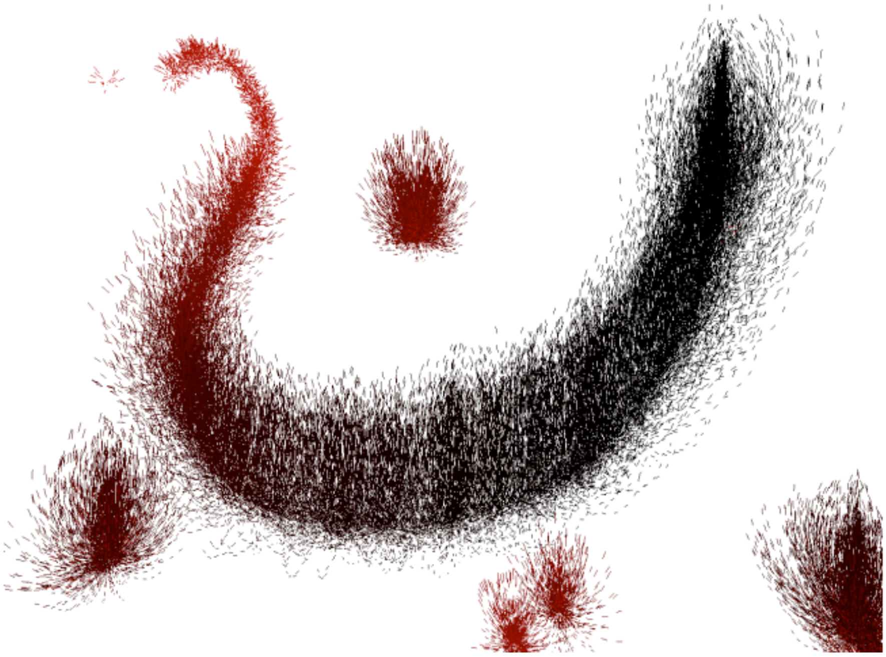


Figure 36: An image generated with the *Spray* tool.

In order to have more control over this algorithm I have set default values for some of the particles characteristics allowing gestural control over three kinds of parameters (CP in Figure 37):

- Colour parameters,
- Position parameters (X, Y, Z) and
- Particles duration (kill time in milliseconds) parameter.

Gesture Measure	*Spray Tool Parameter*	*PAF Synthesis Parameter*
X (horizontal) position	Spatial Localization X axis	Center Frequency
Y (vertical) position	Spatial Localization Y axis	Fundamental & Center Frequency
Local Pressure	Kill time & Brightness	Bandwidth, Vibrato & Amplitude
Duration	Duration	Duration

Table 7: A mapping example Stylus/tablet – *Spray Tool* - PAF (sound) synthesis.

As a result of the mapping proposed in Table 7 diverse kinds of visual forms can be created through variations in colour brightness and the generation of different sizes of “visual clouds” as the user presses harder with the stylus. These “visual clouds” move in different trajectories as a real-time representation of the manipulation of the stylus across the *Wacom* tablet. The GEM window shows these forms until the buffer is emptied by pressing the middle button of the stylus or by clicking the *restart* button in *Miró’s* main control panel.

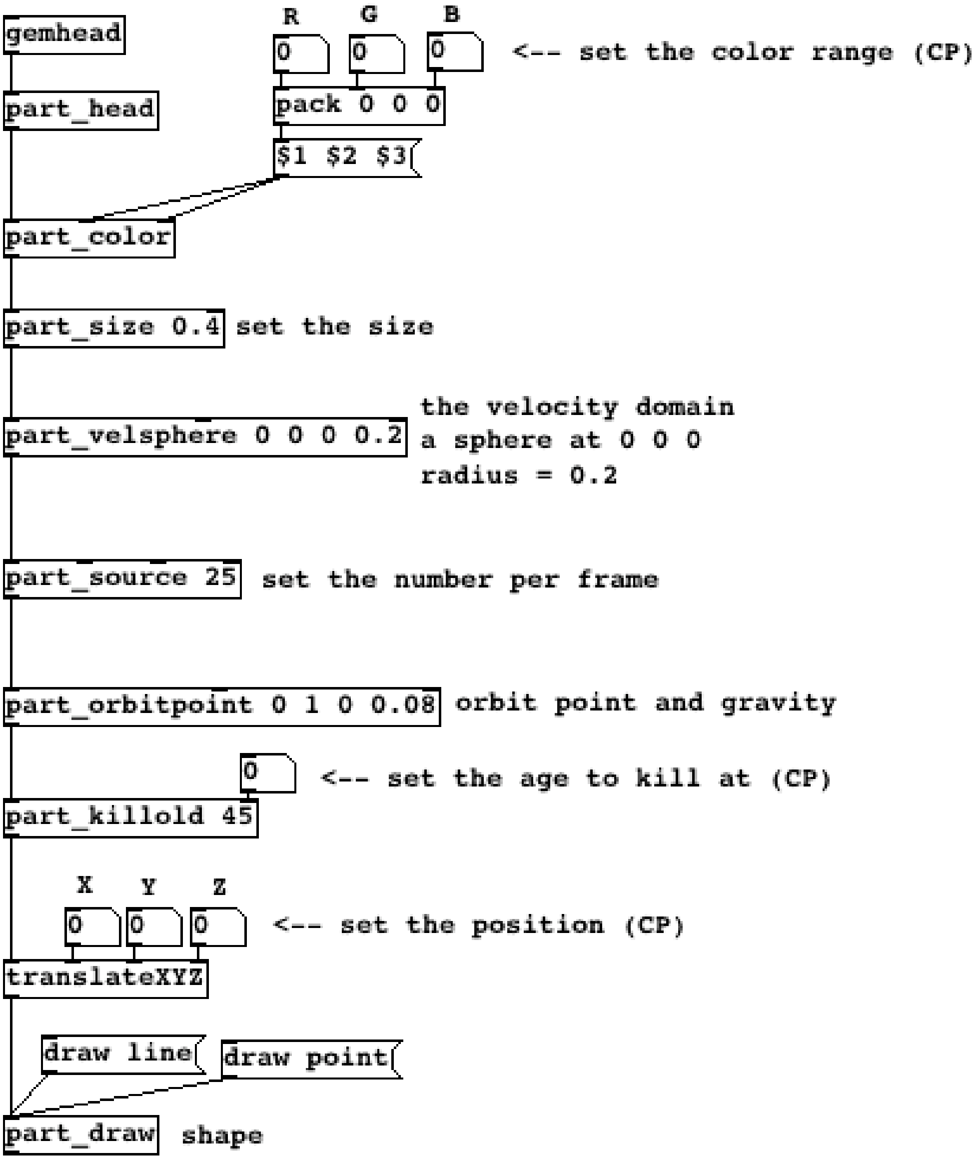


Figure 37: ***Miró’s Spray*** **tool synthesis algorithm.**

5.4.1.3The Fountain Tool

The algorithm that defines this model, shown in Figure 39, is also based on GEM's particle generation objects.

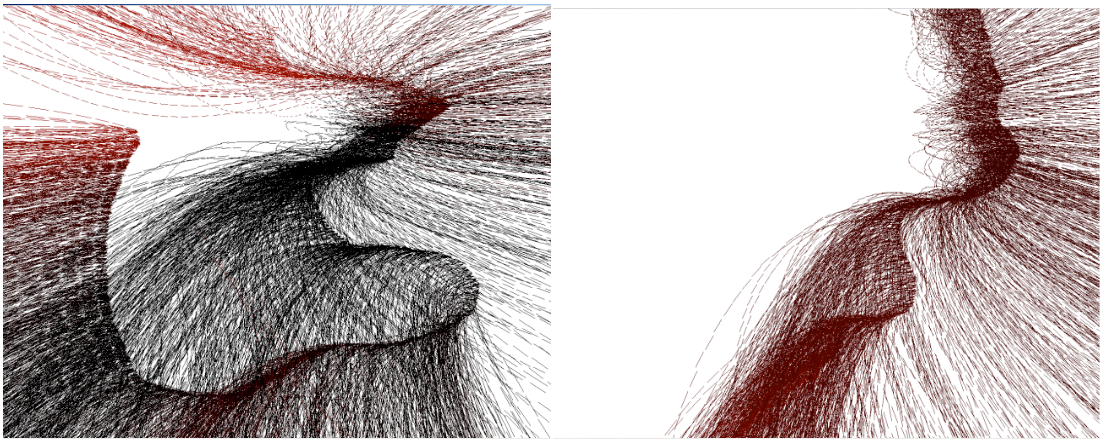


Figure 38: Images generated with the *Fountain* tool.

The *fountain-like* appearance of the images created with this tool is achieved by specifying various particle characteristics such as:

- Particle size.
- Velocity sphere: sets a sphere with a specified radius and midpoint to be the velocity-domain of newly emitted particles within the system.
- Particle source: the initial (variable) argument gives the number of particles that are emitted at each rendering-frame.
- Particle gravity: sets the gravity-vector (x, y, z) of the particle-system. No matter in which direction particles are emitted in the end, they have to follow the gravity.
- Kill old: kill all particles that are older than the kill time.
- Particle draw: finally draws a particle system set up with the other objects. For the *fountain* tool I am using the line drawing mode.

In order to have more control over this algorithm I have set default values for some of the particles characteristics allowing gestural control over four kinds of parameters (CP in Figure 39):

- Colour parameters,
- Number of particles emitted,
- Gravity-vector and
- Position parameters (*X, Y, Z*).

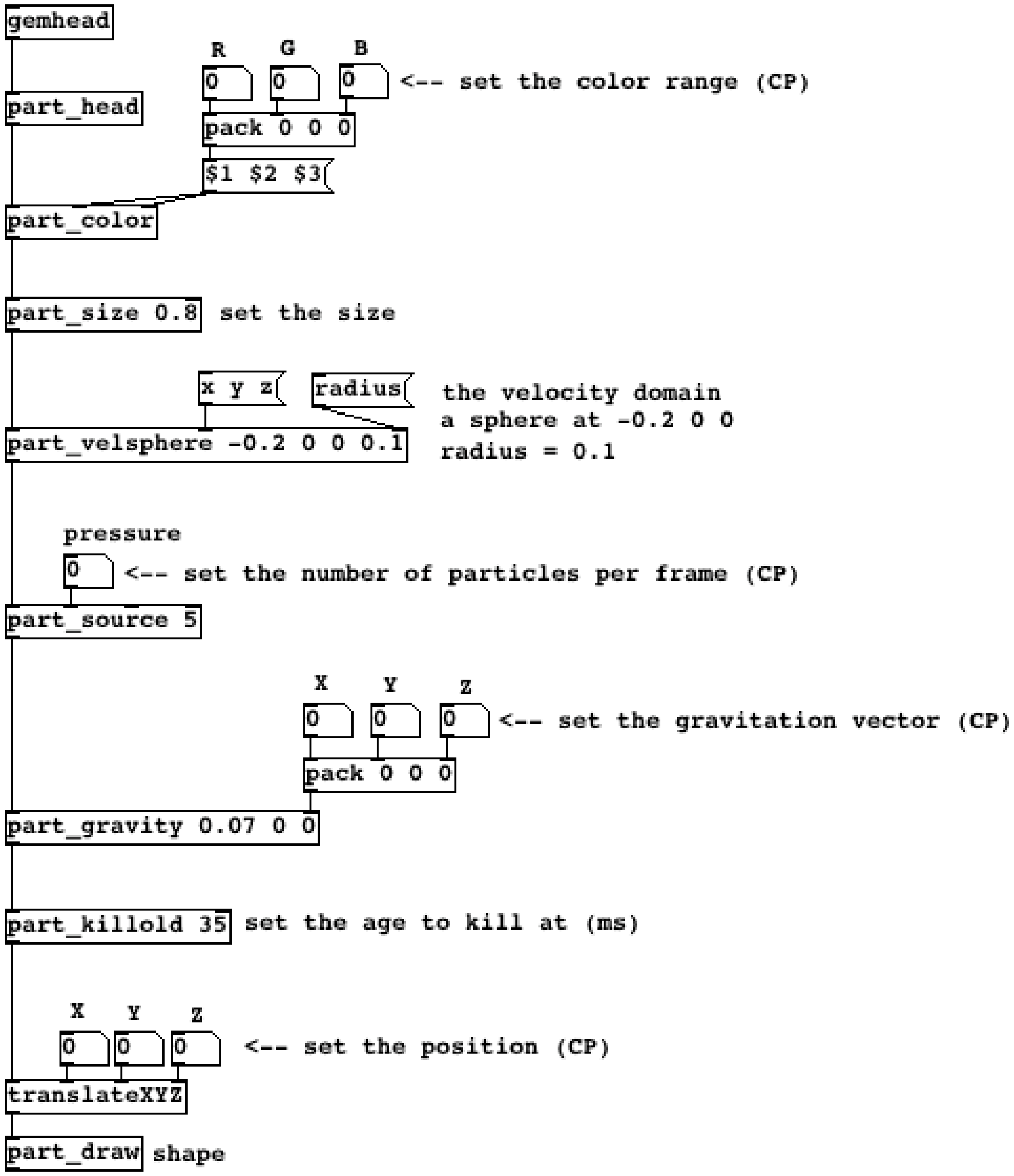


Figure 39: ***Miró's Fountain*** **tool synthesis algorithm.**

As a result of the mapping proposed in Table 8 interesting visual forms (see Figure 38) can be created by controlling colour brightness and number of particles emitted at each rendering-frame as the user presses with the stylus. These forms follow different trajectories as a real-time representation of the manipulation of the stylus across the *Wacom* tablet. The GEM window shows the sequence of images until the buffer is emptied by pressing the middle button of the stylus or by clicking the *restart* button in *Miró's* main control panel.

Gesture Measure	*Fountain Tool Parameter*	*PAF Synthesis Parameter*
X (horizontal) position	Spatial Localization X axis & Gravity	Center Frequency
Y (vertical) position	Spatial Localization Y axis & Gravity	Fundamental & Center Frequency
Local Pressure	Number of Particles & Brightness	Bandwidth, Vibrato & Amplitude
Duration	Duration	Duration

Table 8: A mapping example Stylus/tablet – *Fountain Tool* - PAF (sound) synthesis.

5.5 Setup and Control

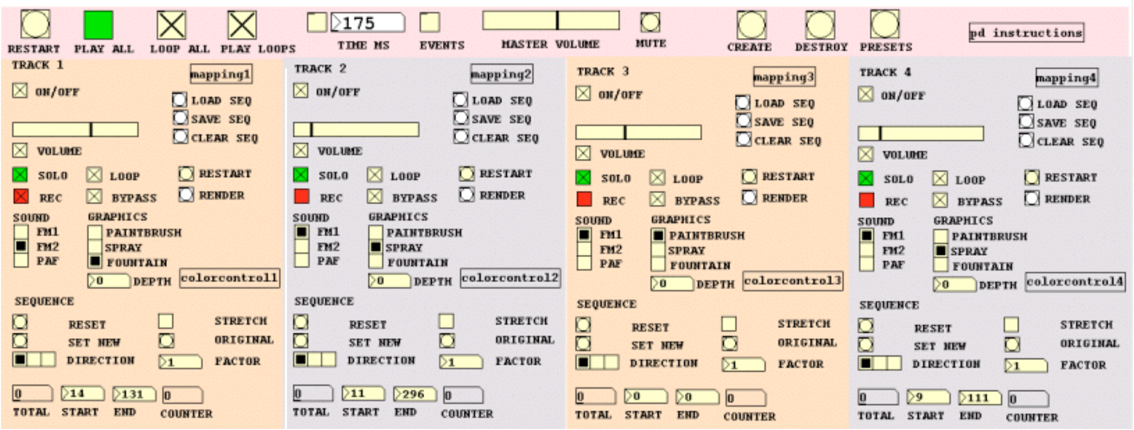


Figure 40: A screenshot of *Miró's* main and tracks control panels.

There are various interaction modalities in *Miró* that mediate the characteristics of its *gesture-image-sound* system. The user can interact with the system at different levels; from the manipulation of the *Wacom* stylus/tablet (instrument manipulation) to generate raw audiovisual material that is captured and recorded in terms of trajectory, force and temporal variations (modulations), to other more abstract or "passive" levels involving playback and organization of events and sequences (score-level, conductor).

There are various control panels in *Miró*, implemented in WIMP-like (Windows, Icons, Menus, Pointing) interfaces by making use of the GUI objects in PD (buttons, sliders, number boxes, etc.). These control panels include: *main control, 4 track controls, rhythm control, colour, mapping, and timelines.*

Miró implements a division of the interaction between the user and the system into two schemas: performance schema and composition schema. These are discussed in the next sections.

5.5.1 Performance Schema

In this schema the process of interacting with the system to generate audiovisual sequences is done by manipulating a *Wacom* stylus/tablet device, for mark-making and/or pointing actions to send control information to the computer. Before this the user needs to setup the system in “rehearsal” mode following the steps represented in Figure 41:

1) Create a visuals window by clicking on the *create* button.
2) Turn on the *bypass* switch in the *track panel*.
3) Turn on the *rec* switch in the *track panel*.
4) Select a sound generator (*FM1, FM2 or PAF*).
5) Select a graphics tool (*Paintbrush, Spray or Fountain*).
6) Set *master volume* and *track volume* at comfortable levels.
7) Test the stylus by pressing and moving it on the tablet.
8) Make as many marks as desire to see and hear the audiovisual results.
9) Press the stylus middle button for cleaning the screen.

Once the user feels confident to record some sequences he/she needs to:

1) Turn off the *bypass* switch and all the following sequences are recorded.
2) Turn off the *rec* switch to stop recording.
3) Click on the *render* button in the *track panel* in order to generate a representation of duration and volume levels in the *timelines control* panel (red arrows in Figure 41).

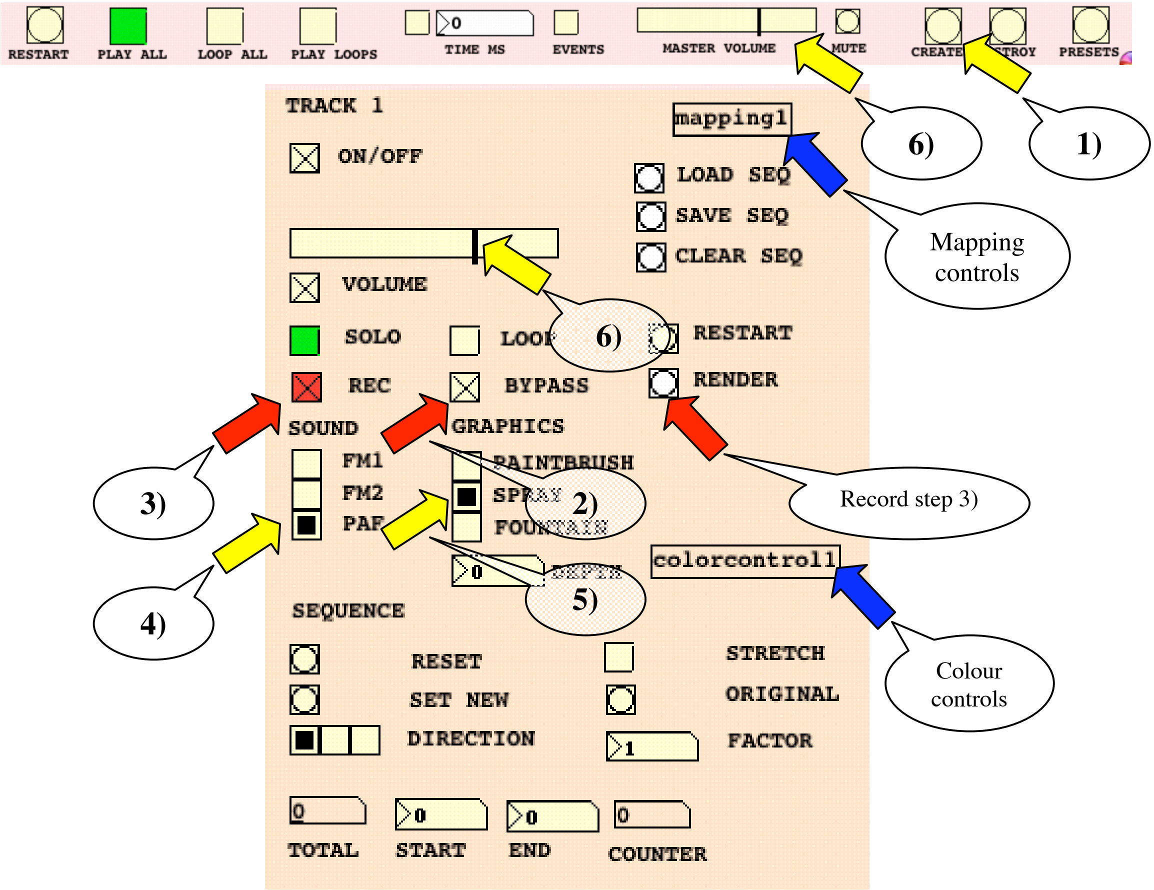


Figure 41: Rehearsal and record set up controls.

The data recorded in a *Miró* sequence includes position (*x* and *y* coordinates), velocity, and pressure taken over its duration. In order to record sequences in other tracks the process described above needs to be repeated.

At this stage many control and synthesis parameters have default values and settings the user can modify to achieve different results. Among these we have the colour and mapping controls that can be opened by clicking on the boxes pointed with blue arrows in Figure 41. The colour controls are sliders to change the value of each colour in the RGB colour space; these are independent for each track.

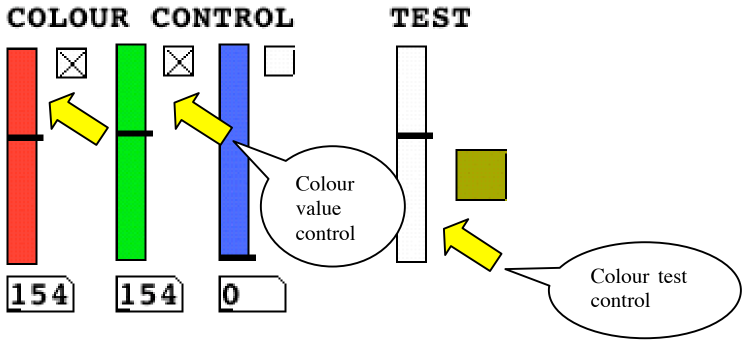


Figure 42: A screenshot of *Miró's colour control* panel.

The user can modify the mapping strategies by turning on and off the switches in the matrix shown in Figure 43. In this way is possible to configure diverse relationships between *gesture, graphics, and sound*.

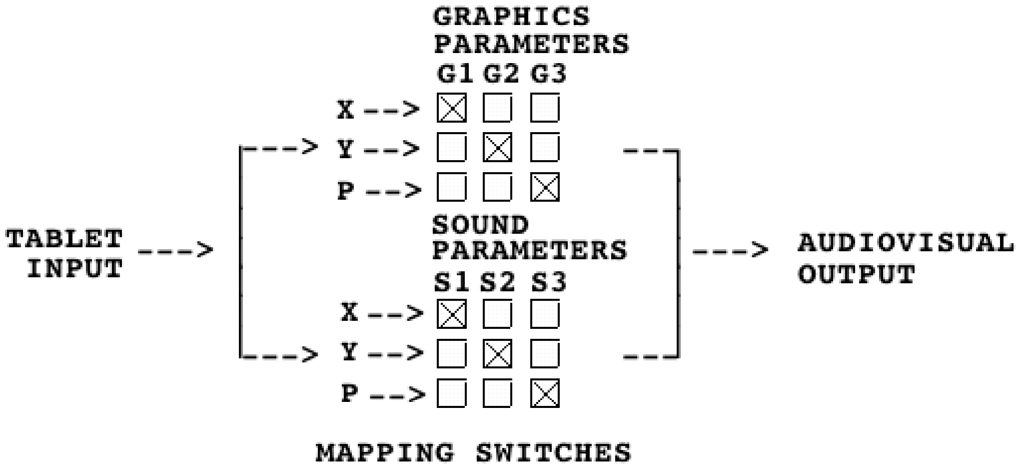


Figure 43: Mapping switches matrix in *Miró.*

Gesture Measure	X (horizontal) position	Y (vertical) position	Local Pressure	Duration
Sound	***S1***	***S2***	***S3***	
Simple FM Synthesis Parameter	Spatial Localization L / R	Carrrier Frequency & Modulation Frequency	Modulation Index & Amplitude	Duration
FM Synthesis Parameter	Modulation Frequency	Modulation Index	Amplitude	Duration
PAF Synthesis Parameter	Center Frequency	Fundamental & Center Frequency	Bandwidth,Vibrato & Amplitude	Duration
Graphics	***G1***	***G2***	***G3***	
Paintbrush Tool Parameter	Spatial Localization X	Spatial Localization Y	Thickness & Brightness	Duration
Spray Tool Parameter	Spatial Localization X	Spatial Localization Y	Kill time & Brightness	Duration
Fountain Tool Parameter	Spatial Localization X & Gravity	Spatial Localization Y & Gravity	Number of Particles & Brightness	Duration

Table 9: Control and synthesis parameters in *Miró*.

In Figure 44 I show the effects of the same gesture using the three different graphics synthesis techniques. The mappings between gesture and graphics used in this example are described in Table 10. This can be easily achieved by switching the radio buttons in the graphics control section of a *Miró's* track control panel shown in Figure 46.

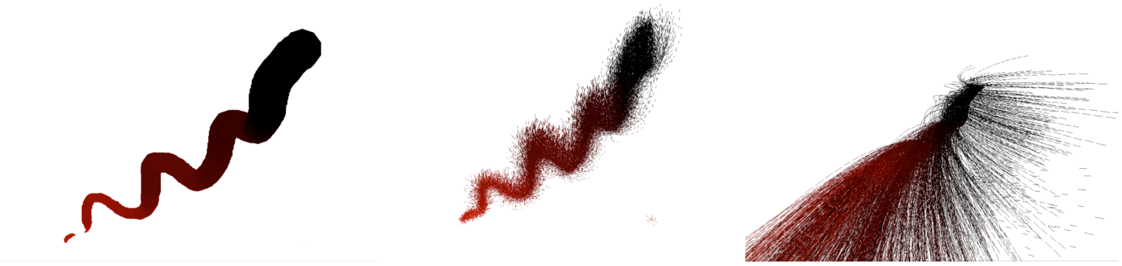

a) **b)** **c)**

Figure 44: Three different visual representations of the same gesture a) *Paintbrush* b) *Spray* c) *Fountain*. See mappings in Table 10.


a) **b)** **c)**

Figure 45: Other visual results from the same gesture in Figure 44 using a different mapping (see Table 11).

Gesture Measure	*Paintbrush Tool Parameter*	*Spray Tool Parameter*	*Fountain Tool Parameter*
X (horizontal) position	Spatial Localization X axis	Spatial Localization X axis	Spatial Localization X axis & Gravity
Y (vertical) position	Spatial Localization Y axis	Spatial Localization Y axis	Spatial Localization Y axis & Gravity
Local Pressure	Thickness & Brightness	Kill time & Brightness	Number of Particles & Brightness
Duration	Duration	Duration	Duration

Table 10: Mappings between gesture and graphics for the visual representations shown in Figure 44.

Figure 45 and Table 11 show the visual change follows changing the original mappings in Figure 44 and Table 10.

Gesture Measure	*Paintbrush Tool Parameter*	*Spray Tool Parameter*	*Fountain Tool Parameter*
Local Pressure	Spatial Localization X axis	Spatial Localization X axis	Spatial Localization X axis & Gravity
X (horizontal) position	Spatial Localization Y axis	Spatial Localization Y axis	Spatial Localization Y axis & Gravity
Y (vertical) position	Thickness & Brightness	Kill time & Brightness	Number of Particles & Brightness
Duration	Duration	Duration	Duration

Table 11: Mappings (changes in bold) between gesture and graphics for the visual representations shown in Figure 45.

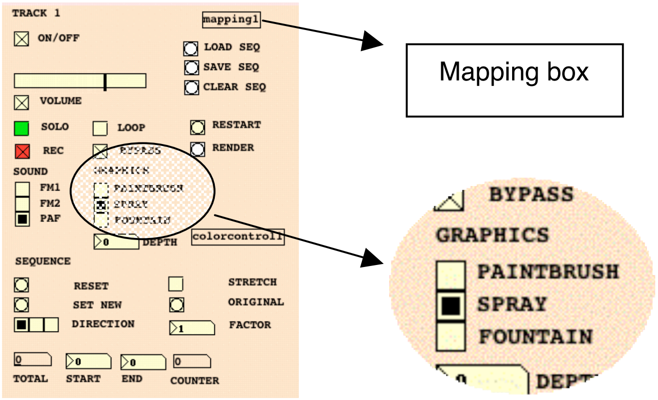


Figure 46: Radio buttons (and mapping box) for switching the graphics tool in the graphics control section.

5.5.2 Composition Schema

In this schema the user can organize and re-organize stored events by setting the correlations of different sequences in terms of synchronization, audio levels, and visual depth (back or front).

It is possible to playback the sequences in three ways:

a) Synchronizing them to a common metronome that triggers each section according to its position on a timeline.

b) According to its own period by creating loops.

c) By creating rhythm patterns in the rhythm control panel.

For the first method (a) I have a two-dimensional representation (Figure 47) of the duration, audio level and playback starting point for each sequence. In this way I have a multitrack-like display with various diagrams or timelines and a common clock (activated in the main control panel) and can make variations in different aspects by modifying these diagrams.

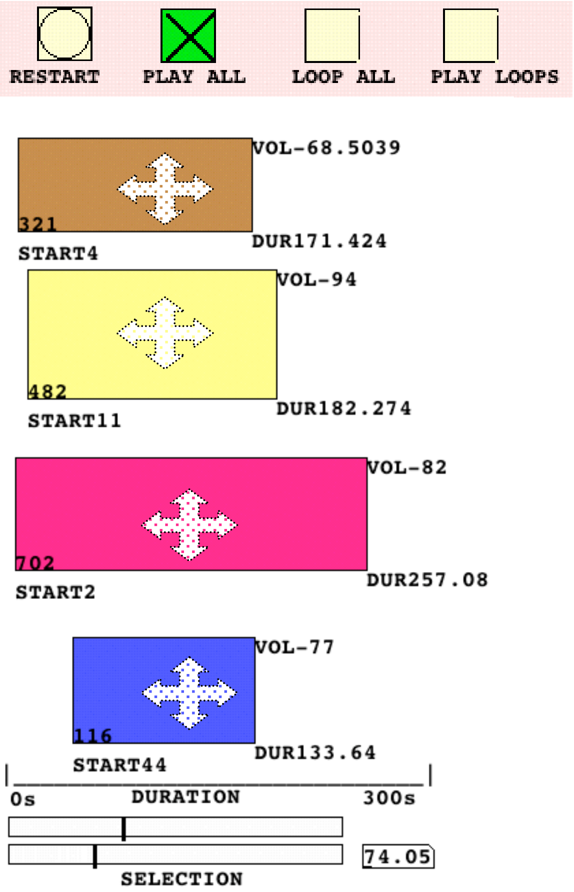


Figure 47: Timelines view in *Miró*.

For instance, it is possible to stretch and shrink the duration of each track, change starting points and audio levels by modifying (height and length) and dragging the shapes that represent the sequences. This method has been implemented using the experimental facility provided in PD for defining and accessing data structures. *"To accomplish this PD introduces a graphical data structure, somewhat like a data structure out of the C programming language, but with a facility for attaching shapes and colors to the data, so that the user* *can visualize and/or edit it. The data itself can be edited from scratch or can be imported from files, generated algorithmically, or derived from analyses of incoming sounds or other data streams"* (PD documentation).

The second method of playback (b) allows the user to improvise with the recorded sequences by creating loops and changing "on the fly" different controls of the track panels such as slot selection, stretch factor, visual depth, colour, playback direction and the graphics and sound synthesizers.

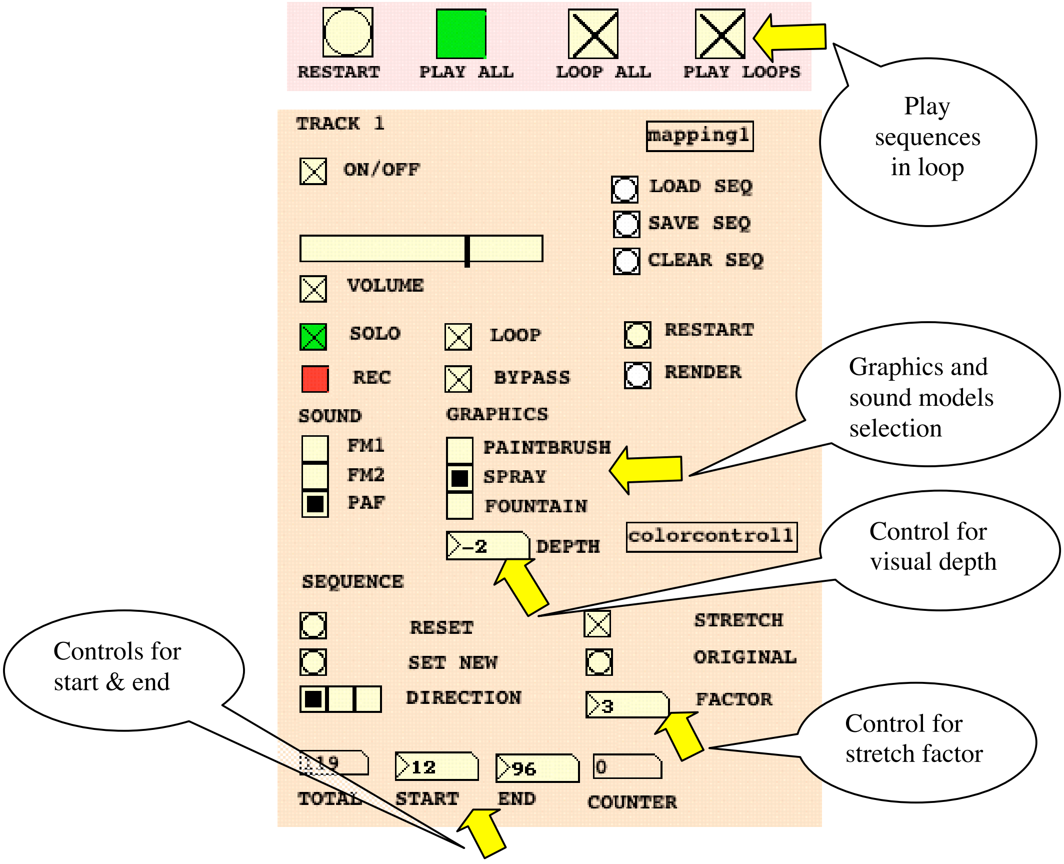


Figure 48: Controls for creating and modifying loops in *Miró.*

These modifications can also be done off-line. For this purpose separated control panels for each track or sequence and a set of general controls have been implemented.

The third method (c) allows the user to create rhythm patterns. To accomplish this I am using the *monorhythm* object created by Mark Williamson for PD. The *monorhythm* object is given a time interval and a rhythm pattern. It divides the interval into the number of beats in the pattern and outputs bangs (trigger messages) as defined by the pattern. The performance time of the entire pattern is given by the time interval. The pattern consists of the symbols “0” and “1” where “0” is a rest (produces no output) and “1” is a beat (produces a bang). In order to synchronise multiple *monorhythm* objects there is a *sync* outlet that does a bang at the start of every bar, which can be fed to a second (or Nth) *monorhythm* objects creating polyrhythms (Figure 49).

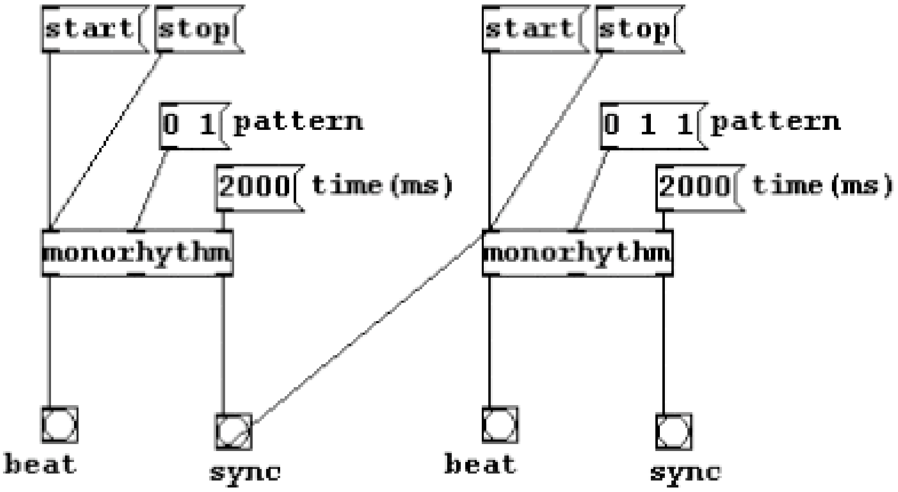
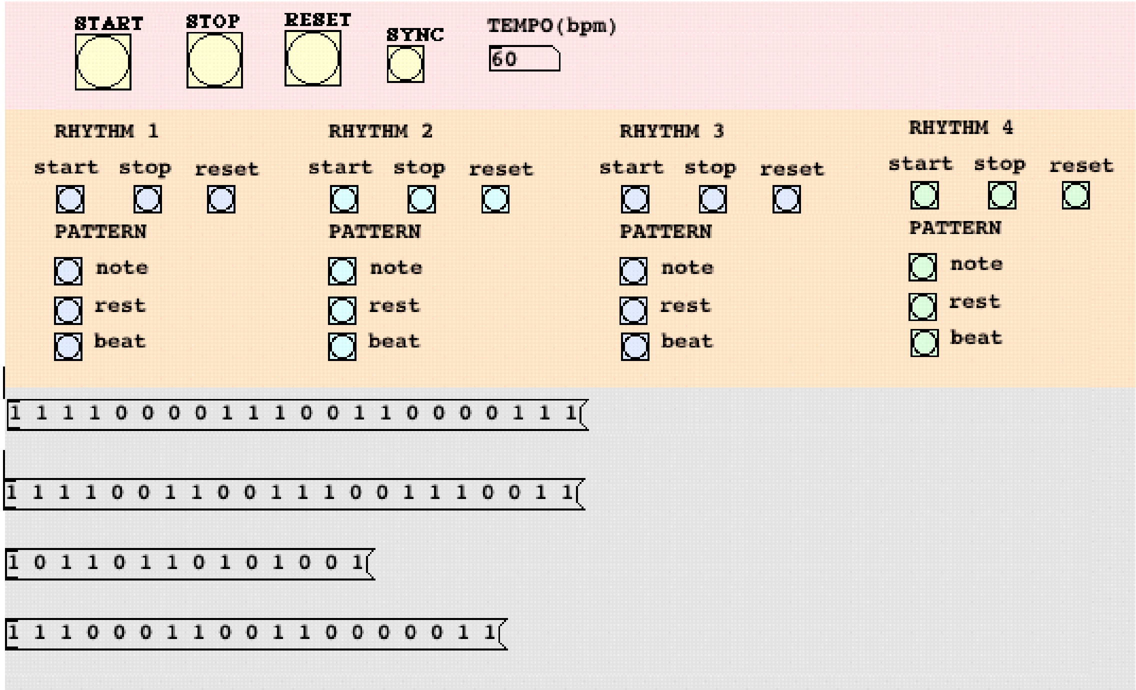


Figure 49: An example of a polyrhythm (2 against 3) built with 2 *monorhythm* objects in PD.

In *Miró* I have implemented a rhythm control panel, shown in Figure 50, which permits the generation of four different patterns that correlate with the four *Miró* audiovisual tracks. In this way the user can create complex polyrhythms by adding beats (notes) and rests in each pattern. These can be created while the sequences are running and/or off-line.

The features of the system presented in this section allow the user to address fundamental issues to be considered in an audiovisual composition process such as:

- Generation and synchronization of sonic and visual events.
- Storing musical audiovisual events.
- Adding and/or deleting events.



TEMPO(bpm) 60

```
1 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 1 1 1
1 1 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 1
1 0 1 1 0 1 1 0 1 0 1 0 0 1
1 1 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0 1 1
```

Figure 50: A screen shot of *Miró's* rhythm control panel.

- The organization of the events in a timeline.
- Parallel event organization strategies (various events starting at the same time, tracks, rhythmic grids).
- Modification and/or transformation of sounds (volume, time variations) and images (change colour, brightness, size, visual depth).

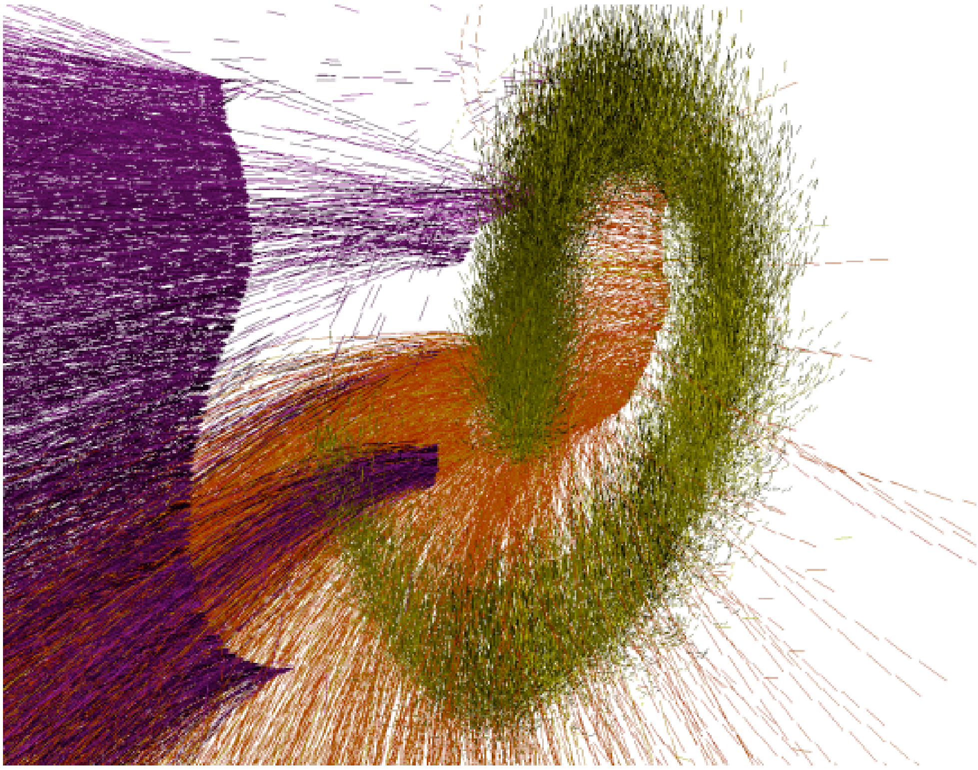


Figure 51: A screenshot of *Miró* in action.

5.6 Summary

In this chapter I have discussed the design of the *Miró* system in terms of its structure, controller, sound and image aspects, set-up and control.

The main aspects of *Miró's* design include the use of gestural control for real-time sound and image synthesis. An important issue within the gestural control is the multi-dimensionality of the hand gestures including pressure, direction and velocity that are changing all the time. I focus on the continuous changes represented by these gestures in order to add expressiveness to the audiovisual outcome. However, there are other ways to perform the audiovisual events or sequences in our system by interacting with the control panels implemented in WIMP-like (Windows, Icons, Menus, Pointing) interfaces. This way the user sometimes acts as a conductor (playing the stored events in a improvisatory way) and sometimes as a performer (creating new events from scratch).

As a result of the design process I realized that if I want to keep the audiovisual specification, realization and expressiveness closely linked, there is in fact a contradiction in doing thorough modifications of the audiovisual characteristics of data recorded from gestural input. Consequently, to change a section I decided it is more appropriate to just re-draw it rather than go through every recorded millisecond. The priorities of the prototype, therefore, are the mapping switching mechanism between *gesture-image-sound* and the temporal organization of recorded sections by playing them back using three methods including: the change of insertion points along a timeline, the generation of loops in improvisatory way and the creation of rhythm patterns. The system also supports various diagrams and control panels to modify different aspects of the stored visual and sonic sections such as colour, paint tool, duration, looping, slot selection and playback direction.

The outcome of interacting with *Miró* is a wide range of expressive images and sounds. These are generated through a flexible and adaptive interface, which permits the use of the system in a number of different scenarios such as: *live* *audiovisual performance, audiovisual composition, abstract animation* and *music therapy.* The use of *Miró* in these scenarios is discussed in chapter 6.

Chapter 6. Analysis of *Miró*

In this chapter I analyse the performance of the *Miró* prototype in terms of its *Structure and Design Aspects, and Interaction Aspects.* In these sections I discuss its strengths and weaknesses and propose ideas for improving future versions of the system. In the *Evaluation* section*,* I assess the *Miró* system according to the set of properties proposed in Chapter 2 as desirable for building an interactive audiovisual system. In the *Scenarios* section, I propose a number of different contexts in which *Miró* can be used. The last section presents some ideas for *future work* and further development of the system.

6.1 Structure and Design Aspects

The system visualizations and sonifications are based on a mixture of perceptual properties of the marks and shapes executed or performed by the user, and the use of a Cartesian grid (*x* and *y* coordinates in the visuals window). These perceptual properties include direction, velocity, shape, colour, and texture. It is also important that the synthesis parameters are mapped to temporal and/or spatial controls from the input device. For example the duration and trajectory of a visual sequence depends on the velocity (time-based) and direction (space-based) of the user's gestures. This approach permits the implementation of alternative control interfaces with a tight relationship between gesture, dynamic image and sound (Levin, 2000). The mapping switching mechanism implemented in *Miró* allows the user to experiment with the relationship between control parameters and synthesis parameters and therefore the system can be used to explore new types of interaction and control.

One of the most useful features of *Miró* is its ability to record and playback temporal, spatial and force (pressure) properties of the user's marks or gestures. *"The advantages of this gesture capture technique are its tremendous capacity to produce lively, organically-animated results, and the exceptionally tight relationship it establishes between the user's input and the system's output"* (Levin, 2000). For instance, the temporal specification is used for the evolution of the audio and animations. Also, the gestural inputs are used as spatial specification for the visuals and sounds and for simultaneously specifying visual shapes, textures and colours and audio intensities, frequencies and timbral variations.

The system uses several different modes of interaction (drawing/performance, record, playback). Consequently, various control panels have been implemented for specific control functions (track controls, rhythm controls, colour, loops, etc). Although this expands the control possibilities, operations, states and products of the system, it also makes the system's operation more difficult to learn and less intuitive. It would be more useful to have control over different functions by using interactive graphics techniques (e.g. drawing and modifying waveforms, envelopes, spatial paths) instead of having control panels of switches and buttons.

Also, the development of more complex PD data structures (PD documentation) would allow editing based on graphics or shapes whose properties can be modified in relation to different aspects of the recorded data contained in the audiovisual sequences. The system's *timelines* control based on PD data structures (section 5.5.2) shows the advantage of attaching shapes and colours to the data for a more intuitive control.

6.2 Interaction Aspects

The first observation I can make after running several informal tests is that *Miró* works well as a real-time sound and graphics generator system controlled by human gestures. To be more specific, if I am using just a single *Miró'* track in "rehearsal" mode (see section 5.5.1) with its correspondent graphics and sound generators, it is possible to perform any number of audiovisual sequences and empty or clean the visuals window at will. The high-resolution and multi-dimensional input from the *Wacom* stylus/tablet makes the control of the system very flexible and accurate. This input device is very sensitive and the system in general offers low latency, and consequently many expressive mappings can be established between the system's input and output. The implementation of three different sound generators and three different graphics generators illustrates the advantages of the system's mapping flexibility, and therefore the variety of its audiovisual outcome.

The visual results of using the mapping presets (described and exemplified in sections 5.3 & 5.4) for the *Paintbrush* and *Spray* tools are very predictable once the performer gets used to the functioning of the system in "performance" mode (section 5.5.1). This is because there is a strong intuitive association between drawing gestures and quality of the marks or strokes that can be achieved by using "real" drawing or painting tools such as a paintbrush, pen or spray can. The outcome of using the *Fountain* tool is less predictable but at the same time this has its own attractions if a degree of uncertainty is desirable.

The sounds produced by interacting with the system can be unpredictable and awkward for a novice user, i.e. someone not used to sound synthesis control and/or *Wacom* stylus manipulation. This is due to the direct low-level control of sound properties and the variety of results the system can generate with each of the three synthesis methods (two FM models and the PAF model). However, as mentioned in section 5.3.1.3, the *PAF's* spectral evolution is relatively simpler compared to the output of the FM synthesis, and it is therefore easier to achieve desired sonic results. The *gesture-image-sound* mapping strategies are more complex and can seem more arbitrary and less intuitive than the *gesture-image* ones. Therefore more time and experimentation is needed for the users to become comfortable with the system's response to their gestures.

In terms of editing and compositional functions *Miró* supports direct events organization approaches (section 5.5.2) in relation to a master timeline. In this way the user can modify the temporal relationships between sequences by changing starting points and durations. Although the system facilitates making these modifications "on the fly", in an improvisatory way when the sequences are playing as well as in off-line mode; there is currently too much decision-making left to the user. It is left entirely up to the composer/performer to compose the sequences in such a way so as to realise their musical and aesthetic goals. This openness is essential, but it is recognised that while a total-control strategy is agreeable for experienced composers, that others might have some trouble building and coordinating these structures, and consequently may loose interest. The rhythm control panel facilitates the creation of rhythm patterns but it is still a very simple prototype. Therefore, the system would benefit from more sophisticated organizational structures in which the current open-control structure can be embedded.

6.3 Evaluation

In Chapter 3, I examined various systems in terms of the properties that are considered desirable for building an interactive audiovisual system (see section 2.2). I will now evaluate *Miró* in terms of these properties, which include: *Real-time (improvisatory) performance capabilities for the creation of images and sound, Compositional structures including events organization and modification, Expressiveness, Mapping flexibility between image and sound, Modifiers, effects and filtering for audio and image, and Learnability.*

6.3.1 Real-time (improvisatory) performance capabilities for the creation of images and sound

As I stated in section 6.2, *Miró* has the capacity to create images and sounds from scratch by taking advantage of the multi-dimensionality, variety and nuances of the user's hand gestures associated with drawing. These properties are similar to some of the systems examined in Chapter 3, especially Levin's *Loom* in which a tight relationship is established between sonic and visual events. However, the implementation of various sound and graphic generators within the same structure in *Miró* permits the performance of a more flexible and more diverse set of audiovisual events with different characteristics.

6.3.2 Compositional structures: events organization and modification

Miró can organize recorded events or sequences using various methods. These include organization of sequences in four parallel tracks along a timeline, the variation of starting and ending points within the sequences and the creation of rhythm patterns. Different visual and sonic aspects can be modified such as duration, audio levels, synthesis model, visual depth, and colour among others.

Though this allows the user to create short compositions based mainly on his/her own decisions, more sophisticated or rather higher level structures are needed.

6.3.3 Expressiveness

The adaptation of the *Wacom* stylus/tablet for gestural input allows detailed control over synthesis parameters based on the temporal and spatial characteristics of the gestures made. This controller can capture a wide range of gestures and nuances from the user's performance. I have used those related to movements across the *x* and *y* axes, and pressure for making continuous modifications in the synthesis parameters. In this way it is possible to record and playback audiovisual sequences based on human gestures, which results in expressive and organic animations and a variety of sound timbres.

6.3.4 Mapping flexibility between image and sound

Several graphics and sound synthesis models have been implemented in *Miró*, these are connected to the gestural input through a mapping switching mechanism controlled by the user. In this way the user can experiment with different configurations between aural and visual dimensions in response to his/her gestures. Some of these mappings can seem less arbitrary or more intuitive than others but the user is free to choose the level of "arbitrariness" of the mappings in order to feel comfortable with the audiovisual feedback from his/her gestures. The system uses default configurations or presets that allow the novices to play with the system before experimenting with the mappings.

The modularity of PD architecture easily permits the addition of new synthesis models for both sound and graphics. There is a growing community of PD developers (PD iem website) that supports the creation of new PD objects and patches the user is free to use in any PD application such as *Miró*.

6.3.5 Modifiers, effects and filtering for audio and image

In the present development system prototype the user can modify diverse aspects of the audiovisual sequences including slot selection, stretch factor, visual depth, colour, playback direction and the graphics and sound synthesizers. It is also possible to change duration, audio level and playback starting point for each sequence. However, all these modifications maintain the signature of the original gestural input, which can be guessed through the evolution of the animations and sounds. In order to achieve more sophisticated results, sonic and visual effects and/or filters (e.g. reverb, delay, blur) can be implemented as a part of future development.

6.3.6 Learnability

Due to the fact that *Miró* offers several modes of interaction there are different learning curves associated with each one. For instance, in performance mode it seems easier to learn how to perform the graphics than the sounds. This is most obviously attributed to the straightforward association of drawing gestures to visual results. On the other hand, “drawing sounds” and in particular when sounds are synthesized abstract sounds, can be a very odd activity. Although very interesting in terms of control possibilities and variety of results, it requires more practice time to master. Also, a user may find it easier to control some synthesis models than others. In playback mode various operations can be performed from control panels over different aspects and these need user instruction. This definitely makes the system more difficult to learn. However, after a few tries the user can generally start to produce basic compositions. As proposed before in this chapter, it would be useful to take advantage of the drawing gestures for controlling different aspects of the interaction in a more intuitive way.

In order to complement my own evaluation of *Miró* I asked to naïve users to experiment with the system. These were provided with a questionnaire and asked to explore the system and record their responses. The main issues that I have isolated from their experimentation are:

- The manipulation of a stylus or pen is familiar to almost everyone. It facilitated the control of the system and because of the haptic feedback it allowed the users to focus their view on the screen. However, some of the subjects suggested that the implementation of a pointer on the screen might be useful.
- The link between image and sound was important to facilitate the operation of the system and it encouraged experimentation.
- The *Paintbrush* tool was the easiest to control.
- The mapping between stylus position along the *x* axis and the sound panning (left and right) was very useful to hear what effect the users' gestures had on the sound.
- The PAF sound synthesis model was the most expressive and engaging in terms of control and sonic results.
- The manipulation of the stylus makes it easy to create slur and/or glissando effects on the sounds.
- Controlling playback through the control panels was not immediately obvious.
- Due to timescales the timeline bars offer more intuitive ways to control the playback of sequences, but in terms of usability they need to be improved.

The recorded responses of the users can be found in Appendix A.

6.4 Scenarios

The control flexibility and diversity of results generated in *Miró* makes it suitable for the use in a number of different scenarios. Here I consider: live audiovisual performance, audiovisual composition, abstract animation and music therapy.

6.4.1 Live Audiovisual Performance

The real-time features of *Miró* (some of them mentioned in the performance schema in section 5.5.1) make the system suitable for live audiovisual performance set-ups. It would be interesting to explore the possibilities of different formats (e.g. duo, trio) where more than one performer plays *Miró.* In this way, it would be possible to experiment with scored and improvised pieces of dynamic visuals and sounds. This experimentation could yield interesting relationships between the roles of performer and conductor.

6.4.2 Audiovisual Composition

Some possibilities of using *Miró* as an audiovisual composition environment have been proposed within the composition schema (see section 5.5.2). However, other composition methods are waiting to be discovered by taking advantage of *Miró's* open interface that allows the user to sketch and think about different ideas and approaches to the aural and visual domains. For instance, some users could focus on the sounds as a main goal of expression, while others could focus on the images. The visual outcome of interacting with *Miró* can also works as an experimental graphic score where the visual specification and sound realization are closely linked.

6.4.3 Abstract Animation

The use of *Miró* with focus on visual aesthetic goals fits in with what some artists and organizations, such as *Iota* (*Iota*, 2004) directed by Larry Cuba, Roberta Friedman and Sara Petty, call *the art of light and movement*. Following this approach *Miró* can works as a system for creating graphic cinematic expression given its capacity for playing “coloured rhythms” with diverse shapes and textures. However, this visual expressiveness can always be expanded with tightly coupled sounds.

6.4.4 Music Therapy

Pressing (1997) proposed that interactive music systems *“may also be designed to have an adaptive control relationship between gesture/posture and sound, which can usefully accommodate individual differences…Such adaptiveness has therapeutic implications for the disabled”* (Pressing, 1997). The adaptability of *Miró's* control features showed its advantages in this kind of context.

During November 2003 I went to the Brothers of Charity Residential Centre in Bawnmore, Limerick-Ireland to set up my prototype in the music therapy room, and evaluated its usefulness for disabled performers with various degrees of disability ranging from cerebral palsy and Down's syndrome to general learning disability. The prototype, based on MIDI, was a simplified version of *Miró*. I used a piano MIDI sound and the control was over the pitch in the “Y” axis and over the velocity with the pressure variations.

It was very interesting to observe the different approaches used to control the stylus for drawing and playing sounds. Some of the performers used the screen as a guide for their movements while others looked at the controller all the time. There were different patterns of movements such as circles, squares, dots, up and down. Some of the performers applied different pressures in order to get different colour intensities and sound levels.

One of the members of the *Bawnmore Ensemble* played the instrument in the *Common Ground* interactive piece, collaboration between Mikael Fernstrom and Mícheál Ó Súilleabháin, in the *SIONA* festival in Limerick, Ireland in November 2003.

6.5 Future Work

There are various aspects of *Miró* that can be improved. One of the most important developments could be the refinement of the synthesis algorithms and/or the implementation of new ones. Also, the introduction of a palette of effects for both sound an image could be useful. These improvements would expand the expressive range of the system and at the same time would produce more attractive and enjoyable sounds and graphics.

Another aspect for experimentation is the use of other kinds of controllers and/or control strategies. A two-handed interface would permit control over more parameters and/or more detailed control over the existing ones. For instance, having an FM synthesis model, the user could control the carrier and modulation frequencies with one hand (by using a *Wacom* stylus) and the modulation index with the other (e.g. by using pressure sensors, sliders, buttons).

The implementation of more sophisticated structures that can assist the user in the composition process is an important aspect for future work. In this way, as proposed by Oppenheim (1992), *"by having a large palette of tools that overlap in functionality the composers have a better chance at finding a tool that best suites their individual way of thinking"* (Oppenheim, 1992). Also, the use of more interactive graphics techniques (e.g. operations on waveforms or geometric shapes) for editing different aspects of the audiovisual sequences would make the system easier to operate.

Different kinds of setups that allow the collaboration of several users simultaneously performing the system in different computers would permit livelier performances. I have also proposed some scenarios in which *Miró* could be used (see section 6.4).

Chapter 7. Conclusion

Through the design of the *Miró* prototype I have found that the *simultaneous control and generation of sounds and images* at a low level, in an audiovisual system, makes it possible to achieve more expressive results than by modifying predefined shapes and pre-recorded sounds. This approach develops the *Painterly Interfaces* control metaphor (Levin, 2000) in *dynamic visual feedback* that reproduces spatial and temporal dimensions of the performer's gestures, and permits the production of variety of expressive sonic and visual forms. This visual feedback is based on the idea that a timeline unfolds as the user performs gestural marks that are recorded and "re-animated" when played back.

The integration in the *Miró* prototype of diverse visual representations (three graphics synthesis models) and sonic forms (three sound synthesis models) controlled by a gestural input, offers an interesting alternative to current systems to allow expressiveness and flexible control of software-based music.

In the *Miró* prototype I have integrated several important features that exist in isolation in other real-time interactive audiovisual systems. These include a multi-dimensional physical controller with high resolution, and a global view of recorded material distributed along a timeline. The implementation of an electronic drawing device (*Wacom* stylus/tablet) allows the user to make freehand recordable drawings or marks that provide both sound specification and visual feedback.

Various levels of control have been implemented in *Miró.* It allows control at a low-level by creating sounds and images from scratch by making continuous variations on synthesis parameters through gestures captured from a *Wacom* stylus/tablet (gestural) input. High-level control over different aspects of recorded audiovisual sequences (up to four parallel sequences) such as starting points, duration, loudness and colour, is possible through timelines, control panels and diagrams.

The PD programming environment with its real-time graphics and audio processing capacity, and modular architecture supported the requirements for building the *Miró* prototype. Due to the modularity of PD's architecture, the addition of new synthesis models to expand the possibilities of the audiovisual output can be easily achieved. However, it has to be said that PD is more suitable for defining patches (algorithms) for low-level control of sound and graphics that respond to the performer's gestures in real-time, than for generating and controlling higher level structures that permit the modification of different aspects of the recorded audiovisual sequences. Therefore, the composition tools implemented in *Miró* need to be improved with further developments that reflect this duality of control strategies.

The openness and flexibility of *Miró's* structure offers the opportunity to experiment with different mappings between gesture, image and sound. *Miró* also offers a variety of tools and controls for creating, editing, modifying and performing audiovisual sequences. With this palette of tools and controls (that are often redundant and overlap in functionality) composers and/or performers have a better chance at finding a strategy that best suites their individual way of thinking for performing and composing audiovisual pieces. In this way it is possible to explore new ways of interaction.

In summary, I have developed a gesture-controlled painterly audiovisual synthesizer with a built-in multitrack sequencer. It can play up to four simultaneous audiovisual sequences in which the synthesis models for both audio and graphics can be switched at any time of the interaction. The control strategies at various levels can be modified to meet the specific needs of different users. The tight relationship established between gesture, dynamic image and synthesized sound permits the creation of variety of expressive audiovisual results.

The creation of a prototype system to facilitate the exploration of the relationships between gesture, image and sound in composition was the primary aim of this research. While the prototype is in many ways incomplete, I believe that the viability of a system that brings together organisational architecture of the traditional sequencer, the dynamism of physical performance and the flexibility of multiple mappings between gesture, image and sound has been established and will allow further developments to be made.

Bibliography

Abbado, A., 1998. *Perceptual Correspondences of Abstract Animation and Synthetic Sound.* http://www.abbado.com/thesis/corpo.html.

Arfib, D., Dudon, J., 2002. *A Digital Emulator of the Photosonic Instrument.* Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02). Dublin, Ireland, May 24-26. Published by the University of Limerick, Department of Computer Science and Information Systems,128-131.

Cadoz, C., 1988. *Instrumental Gesture and Musical Composition*. In Proc. of the 1988 International Computer Music Conference. San Francisco, Calif.: International Computer Music Association, 1–12.

Camurri, A., Trocca, R., Volpe, G., 2002. *Interactive Systems Design: A KANSEI-based Approach.* Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02). Dublin, Ireland, May 24-26. Published by the University of Limerick, Department of Computer Science and Information Systems,155-162.

Choi, I., 1998. *Cognitive Engineering of Gestural Primitives for Multi-Modal Interaction in a Virtual Environment.* In Proc. IEEE International Conference on Systems, Man and Cybernetics (SMC'98), 1101–1106.

Chowning, J., 1973. *The synthesis of complex audio spectra by means of frequency modulation.* Journal of the Audio Engineering Society 21 (7): 526-534.

Coagula website, 2004. http://hem.passagen.se/rasmuse/Coagula.htm

Cook, P., 2002. *Real Sound Synthesis for Interactive Applications*. A.K. Peters, Natick, Massachusetts.

Cook, P., 2001. *Principles for Designing Computer Music Controllers*. In

Proceedings of the NIME Workshop, CHI 2001.

Farbood, M., 2001. *Hyperscore: A New Approach to Interactive, Computer-Generated Music.* Master's thesis, MIT Media Lab.

Farbood, M., Pastzor, E., Jennings, K., 2004. *Hyperscore: A Graphical Sketchpad for Novice Composers.* Published by the IEEE Computer Society.

Franco, E., Griffith, N., Fernstrom, M., 2004. *Issues for Designing a Flexible Expressive Audiovisual System for Real-time Performance and Composition*. In Proceedings of the 2004 International Conference on New Interfaces for Musical Expression - NIME04, Hamamatsu, Japan, June 3-5, 165 -168.

Galeyev, B., 1999. *What is Synaesthesia: Myths and Reality.* Published in "Leonardo Electronic Almanac", v.7, 1999, N 6.
http://prometheus.kai.ru/mif_e.htm

GEM homepage, 2004. http://gem.iem.at/.

Hunt, A., 1999. *Radical User Interfaces for Real-time Musical Control*. PhD Thesis, University of York.

Hyperscore homepage, 2001-2004.
http://web.media.mit.edu/~mary/hyperscore/.

IBM Research website, 1998. http://www.research.ibm.com.

Iota website, 2004. http://www.iotacenter.org.

Iwai, T., 1992. *Music Insects* website.
http://www.iamas.ac.jp/~iwai/artworks/music insects.html.

Ixi Software website, 2004. http://www.ixi-software.net/

Jordà, S., 2002a. *Improvising with Computers: A personal Survey (1989-2001)*. Journal of New Music Research, 31(1), 1-10.

Jordà, S., 2002b. *FMOL: Toward User-Friendly, Sophisticated New Musical Instruments*. Computer Music Journal, Vol. 26, No.3. pp 23-39.

Jordà, S., 2000-2004. *FMOL* homepage: http://www.iua.upf.es/~sergi/FMOL.

Jordà, S., 2003. *Sonigraphical Instruments: From FMOL to the reacTable*.* Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03). Montreal, Canada, May 22-24, 70-76. http://www.music.mcgill.ca/musictech/nime

Levin, G., 2000. *Painterly interfaces for audiovisual performance*. Master's Thesis, Massachusets Institute of Technology, MIT Media Lab.

Levin,G., 2004. *Audiovisual Environment Suite.* http://acg.media.mit.edu/people/golan/aves/.

Lohner, H., 1986. *The UPIC System: A User's Report*. Computer Music Journal, 10(4): 42-49.

Lowengard, H., 1994. *Software-O-Phones: Homemade Software Instruments.* http://www.echonyc.com/~jhhl/software.html

Mathews, M., L. Rossler., 1969. *Graphical language for the scores of computer-generated sounds*. In H. von Foerster and J.Beauchamp, eds. Music by Computers. New York: John Wiley and Sons, 84-114.

MetaSynth website, 2001-2004. http://www.uisoftware.com/PAGES/ms_presentation.html.

Moore, R., 1990.*Elements of Computer Music.* Englewood Cliffs, New Jersey: Prentice-Hall.

Moore, F. R., 1988. *The Disfunctions of MIDI*. Computer Music Journal, 12(1), 19-28.

Mulder, A., 1994. *Virtual Musical Instruments: Accessing the Sound Synthesis Universe.* In Proceddings of the First Brazilian Symposium on Computer Music.

Mulder, A., 1998. *Design of Gestural Constraints Using Virtual Musical Instruments*. PhD thesis, School of Kinesiology, Simon Fraser University, Canada.

Music Sketcher website, 1998. IBM research, http://www.research.ibm.com.

Ng, Kia., 2002. *Interactive Gesture Music Performance Interface.* Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02). Dublin, Ireland, May 24-26. Published by the University of Limerick, Department of Computer Science and Information Systems,183-184.

O'Modhrain, S. *Gestural Control of Computer-Based Musical Instruments.* Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02). Dublin, Ireland, May 24-26. Published by the University of Limerick, Department of Computer Science and Information Systems, 7-9.

Oppenheim, D., 1992. *DMIX-A Multi Faceted Environment for Composing and Performing Computer Music: its Design, Philosophy, and Implementation.* Delphi.

PixelToy website, 2003. LairWare Software, http://www.lairware.com/pixeltoy/.

PD documentation website, 2004. http://www.crca.ucsd.edu/~msp/software.html.

PD iem website, 2004. http://pd.iem.at/.

Pressing, J., 1990. *Cybernetic Issues in Interactive Performance Systems*. Computer Music J., 14(1): 12–25.

Pressing, J., 1997. *Some Perspectives on Performed Sound and Music in Virtual Environments.* Presence, 6(4): 482–503.

Propellerhead Software, 1998. http://www.propellerhead.se .

Puckette, M., 1996. *Formant-based audio synthesis using nonlinear distortion*. JAES 43(1), 40-47.

Puckette, M., 2003. *Theory and Techniques of Electronic Music* (draft). University of California, San Diego.

Rice, P., 1998. *Stretchable music: A graphically rich, interactive composition system.* Master's Thesis, MIT Media Lab.

Risset, J.C., 1985. *Digital Techniques and Sound Structure in Music*. In C. Roads, ed. Composers and the Computer. Madison: A-R Editions, 113-138.

Roads, C., 1996. *Computer Music Tutorial*. MIT Press, Cambridge, Massachusetts.

Rovan, J., Wanderley, M., Dubnov, S., and Depalle, P., 1997. *Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance.* In Proceedings of the Kansei - The Technology of Emotion Workshop, Genova – Italy.

Rowe, R., 1993. *Interactive Music Systems: Machine Listening and Composing*. MIT Press, Cambridge, Massachusetts.

Soundtoys website, 2004. http://www.soundtoys.net

Stanza website, 2004. http://www.stanza.co.uk

UI software homepage: http://www.uisoftware.com

Videodelic homepage: http://www.uisoftware.com

Wacom website: http://www.wacom.com/

Wanderley, M., 2001. *Performer-Instrument Interaction: Applications to Gestural Control of Sound Synthesis.* PHD Thesis, University Paris 6.

Wanderley, M. and Battier, M., editors, 2000. *Trends in Gestural Control of Music.* Ircam – Centre Pompidou.

Wessel, D., Wright, M., Schott, J. *Intimate Musical Control of Computers with a Variety of Controllers and Gesture Mapping Metaphors.* Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02). Dublin, Ireland, May 24-26. Published by the University of Limerick, Department of Computer Science and Information Systems,171-173.

Whitney, J., 1980. *Digital Harmony: On the Complementarity of Music and Visual Art.* Peterborough, NH: McGraw-Hill.

Whitney, J., 1976. *Computational Periodics.* In Leavitt, Ruth, ed. Artist and Computer. New York: Harmony, p.80.

Willats, J., 2002. *The Syntax of Mark and Gesture*.
http://www.lboro.ac.uk/departments/ac/tracey/somag/willats.html

Winkler, T., 1998. *Composing Interactive Music*. Cambridge, MA: MIT Press.

World Book Multimedia Encyclopedia, 2001.Mac OS X Edition.

Xenakis, I., 1986. *Page from Mycenae-Alpha* (1980). Computer Music Journal, 10(4).

Appendices

Appendix A. Experimenting with *Miró.*

The mappings between gestures, graphics and sounds used in the experiments are described in Table 12 and Table 13.

Gesture Measure	*Paintbrush Tool Parameter*	*Spray Tool Parameter*	*Fountain Tool Parameter*
X (horizontal) position	Spatial Localization X axis	Spatial Localization X axis	Spatial Localization X axis & Gravity
Y (vertical) position	Spatial Localization Y axis	Spatial Localization Y axis	Spatial Localization Y axis & Gravity
Local Pressure	Thickness & Brightness	Kill time & Brightness	Number of Particles & Brightness
Duration	Duration	Duration	Duration

Table 12: Mappings between gesture and graphics used for the tests.

Gesture Measure	*Simple FM Synthesis Parameter*	*FM (sawtooth) Synthesis Parameter*	*PAF Synthesis Parameter*
X (horizontal) position	Spatial Localization L / R	Modulation Frequency	Center Frequency
Y (vertical) position	Carrrier Frequency & Modulation Frequency	Modulation Index	Fundamental & Center Frequency
Local Pressure	Modulation Index & Amplitude	Amplitude	Bandwidth, Vibrato & Amplitude
Duration	Duration	Duration	Duration

Table 13: Mappings between gesture and sound used for the tests.

Two subjects comment about their experience of using *Miró:*

Subject 1:

1) What do you think about the method of control?

- Different. Not like a piano, for example, as no fixed keys, three dimensions of control (x, y, pressure).
- Easy to slur/portamento.
- Don't need to look at the pad when playing.
- Familiar to anyone who has used a pen before (almost everyone).

2) What sounds and graphics tools are the easiest to control?
   - *Paintbrush* had most control.
   - *Fountain* was "all over the place".
   - *Spray* seemed to make more difference to the sound with PAF.
   - *FM* synthesis with left and right panning made it easier to relate position to sound.
   - *PAF* was harder to control, especially with pressure, but that made it better. You learned to be more controlled over time.

3) Do you find the system expressive?
   Yes! Not immediately apparent how to control it, so surprise and learning makes sounds appear as if "discovered". The link between picture and sound is important, as you want to move over the whole square, and colour change encourages experimentation.

4) Do you find it expressive?
   Yes! *Fountain* is almost too flexible, but it depends on mood. Maybe you want to be controlled sometimes, vague and messy the next time.

5) What do you think of the possibilities for playing back the sequences (loops, rhythm patterns, timelines)?
   Definitely different to any sequencer I've used before. Performing music in a traditional way is not really possible, which is good if you're not a musician, but bit of challenge if you are! It's a completely different way of controlling patterns and playback, and would probably, like most instruments, get easier to play with practice.

6) In general: what do you like and what don't?
    - The interface (pen) is great.
    - Controlling playback, looping and recording is not immediately obvious, but not difficult.
    - Volume slider controlling timeline bar is cool.
    - Start and end points of loop being changeable is good, but I went too far and confused PD by having start after end.

Subject 2:

1) What do you think about the method of control?
   Graphics and sound together make it much easier to figure out what is going on, and what effect my actions have on the system. It would be useful to have a pointer to track the movements.

2) What sounds and graphics tools are the easiest to control?
    - *FM* (phase modulation): hard to control pitch.
    - *Simple FM:* easy to hear the effect of my actions when I move across left and right.
    - *PAF:* very easy to control and articulate notes with interesting envelopes.
    - *Paintbrush:* easy at first, and doesn't confuse me as much as the others.
    - *Spray & Fountain:* these tend to cover the canvas in paint very quickly, and then you are unsure of what you are drawing.

3) Do you find the system expressive?
   I found the PAF synthesis quite expressive. The effect of moving the pen from left to right was very good.
   I grew bored of the other methods quite quickly.

4) What do you think of the possibilities for playing back the sequences (loops, rhythm patterns, timelines)?

It makes it more fun to play with the system and expand its possibilities. However, it needs more practice time.

5) In general: what do you like and what don't?
    - It would be nice to have effects (FX) that change the images and sounds.
    - It would be useful to have more colour variations across the screen, e.g. map colour to pitch.
    - It would be useful to have more synthesis models.
    - At the beginning it seems that the graphics affect the sounds, i.e. even the same sound is perceived different because of the use of different graphics.

Appendix B. *Miró* Application Features.

Gesture Capture

- Physical interface: Wacom Tablet.
- 3 axes: X, Y and pressure.
- Duration of the gestures (speed, gaps).

Sound

- Three different synthesis algorithms: two variations of FM synthesis and Phase Aligned Formant-PAF.
- Three or more parameters of control in real-time mapped to X, Y and pressure from the physical interface.
- Four tracks.

Graphics

- Three different synthesis algorithms: paintbrush, spray and fountain.
- Three or more parameters of control in real-time mapped to X, Y and pressure from the physical interface.
- Four tracks.

Control Panels

- *General control*: access to the other control panels, general restart and play, general time display in milliseconds, input status.
- *Track control*: mapper, graphics and sound synthesis algorithms control, on/off switch, record, play, solo, loop, audio level, colour, render to timelines, time stretch, save and load, clear.
- *Graphics control*: create and destroy graphics window, window size, background colour, load and clear buffer.
- *Audio master level*: general and track levels.

- *Loop control*: duration of sections (time stretch), starting and ending points, on/off switches, number of events display, time in milliseconds display, general play and loop switches.
- *Timelines*: visual representation of audio level, duration and starting point. Control of duration (stretch) and starting points.
- *Rhythm control*: generate rhythm patterns.

Appendix C. Stills from *Miró* in action

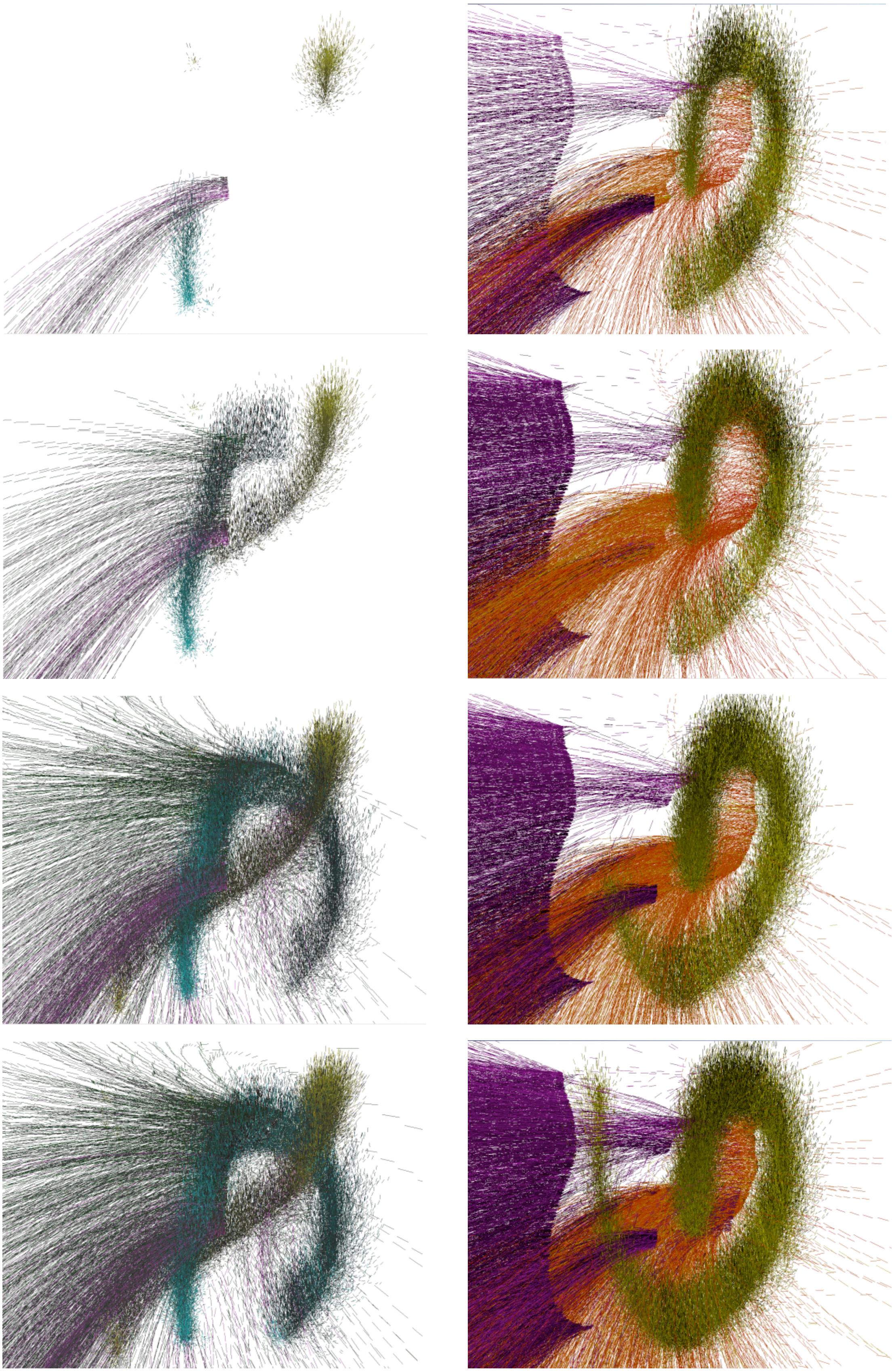


Figure 52: Two sequences generated with Miró.

Appendix D. Application’s Code. Screen Shots – PD Patches.

General control: access to the other control modules including input, graphics window, timelines, tracks, synthesizers, general restart and play and main clock.

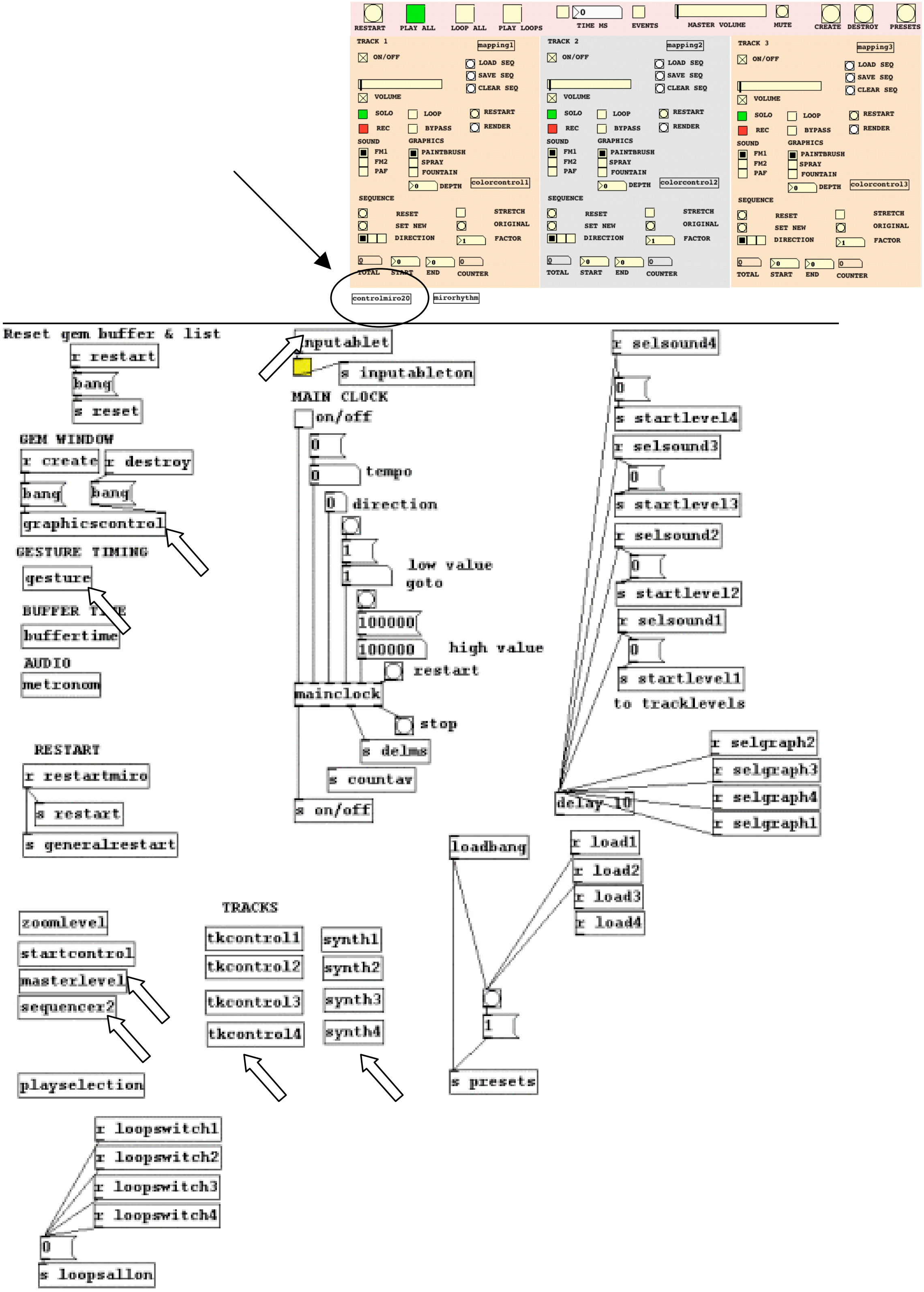


Figure 53: General control module.

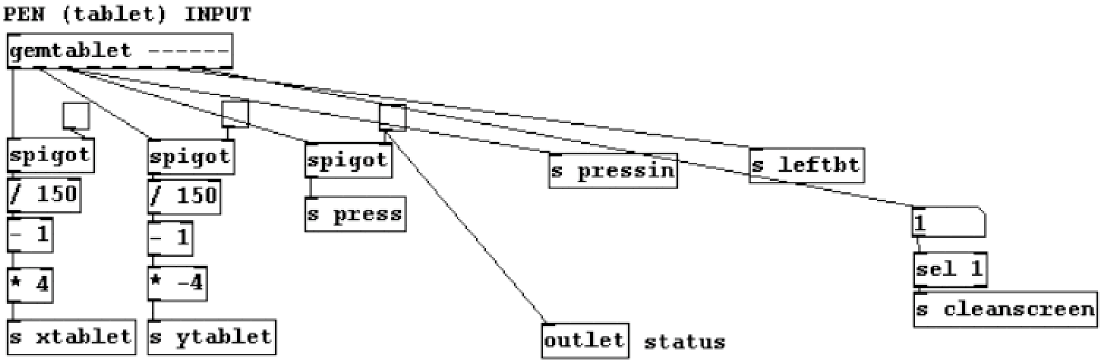


Figure 54: Input module.

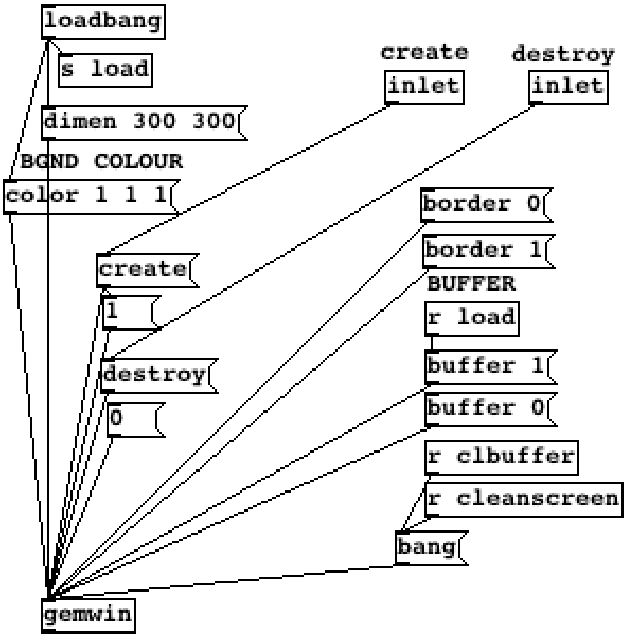


Figure 55: Graphics window control module. Create and destroy the graphics window, window size, background colour, load and clear buffer.

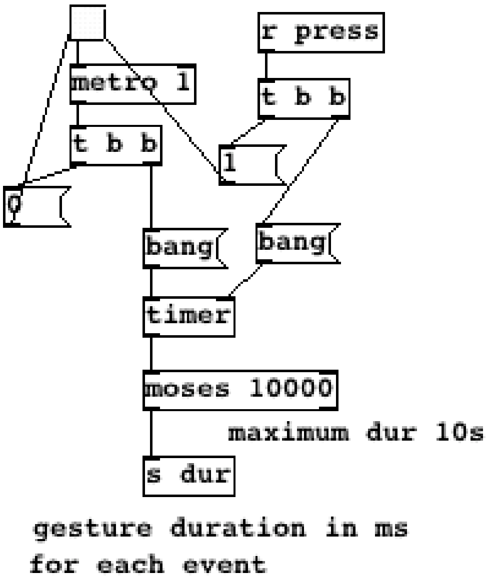


Figure 56: Gesture time module. The important object is *timer*.

Track control: mapper, graphics and sound synthesis algorithms control, on/off switch, record, play, solo, loop, audio level, colour, render to timelines, time stretch, save and load, clear.

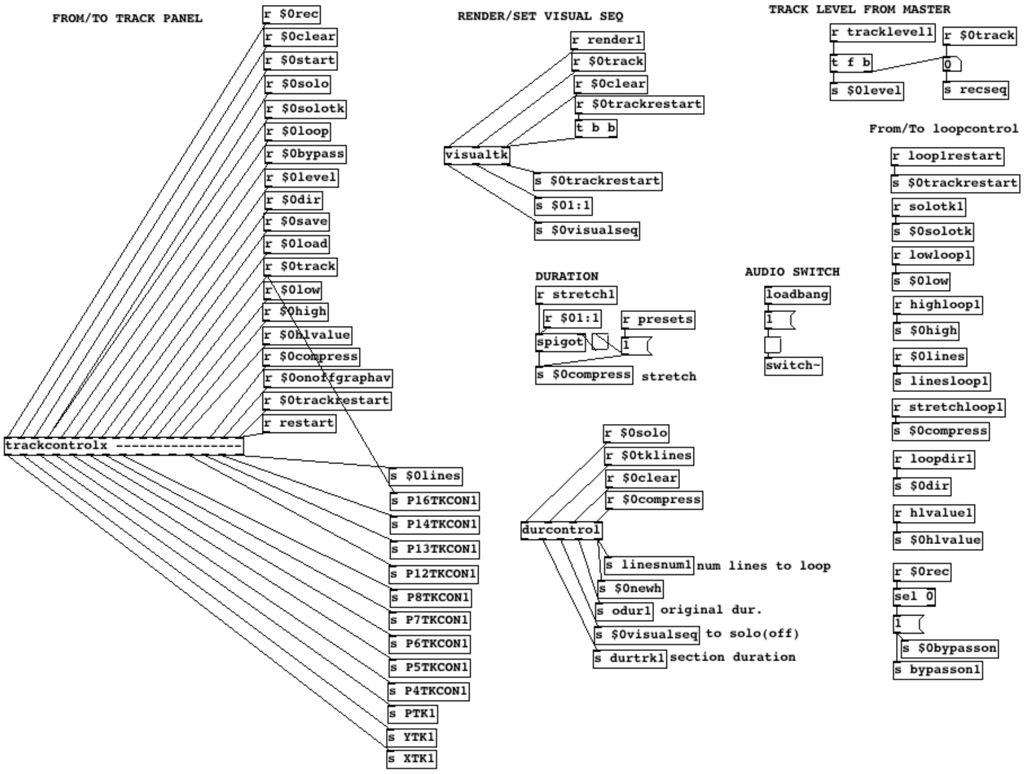


Figure 57: Track control module.

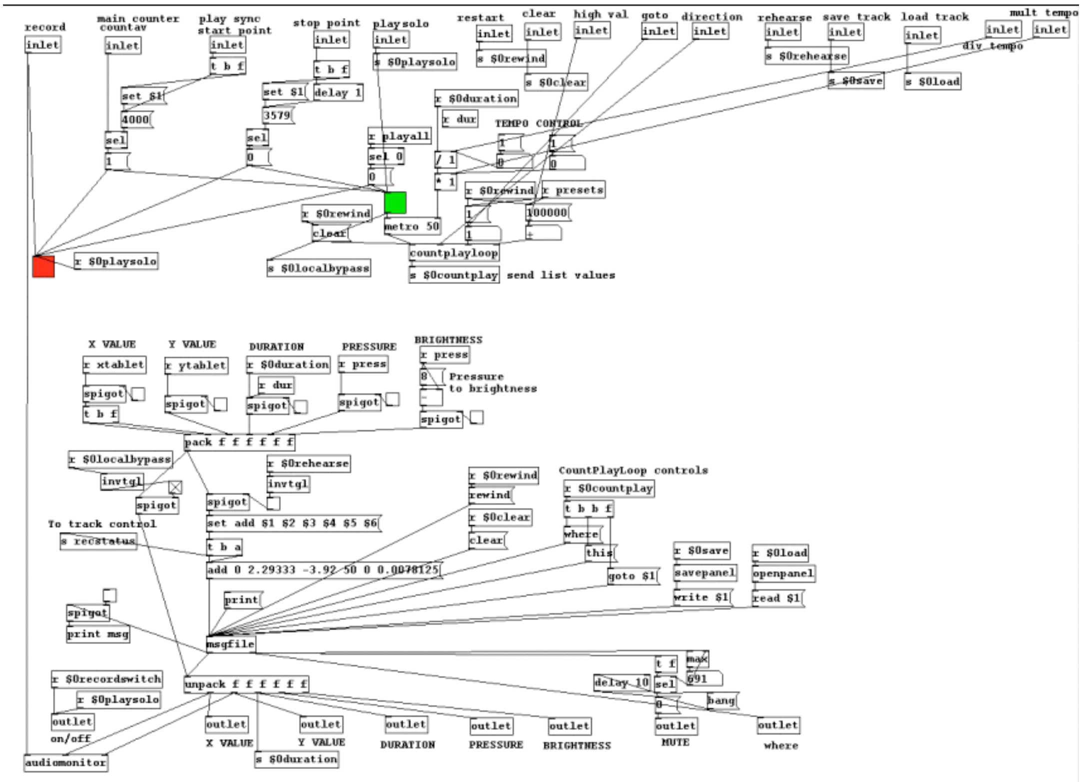


Figure 58: List manager module, the core of this module is the *msgfile* object. The sequences are recorded and played back through this object.

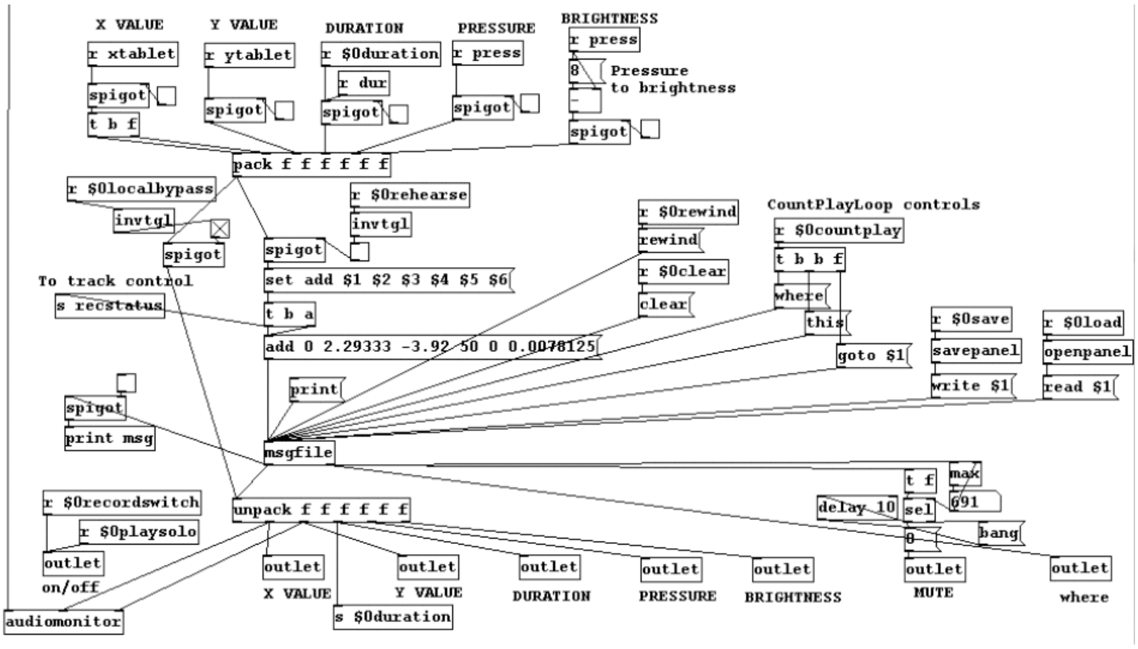


Figure 59: Closer view of the list manager module.

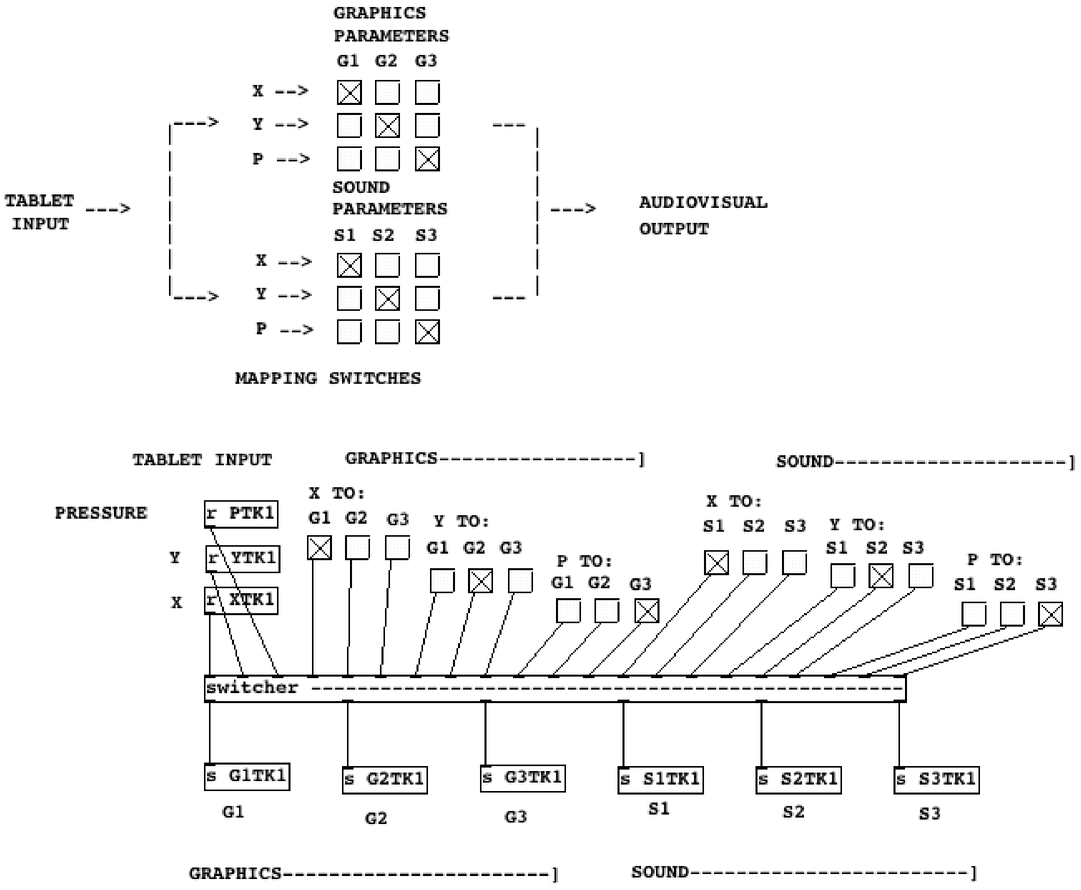


Figure 60: Mapping switcher module.

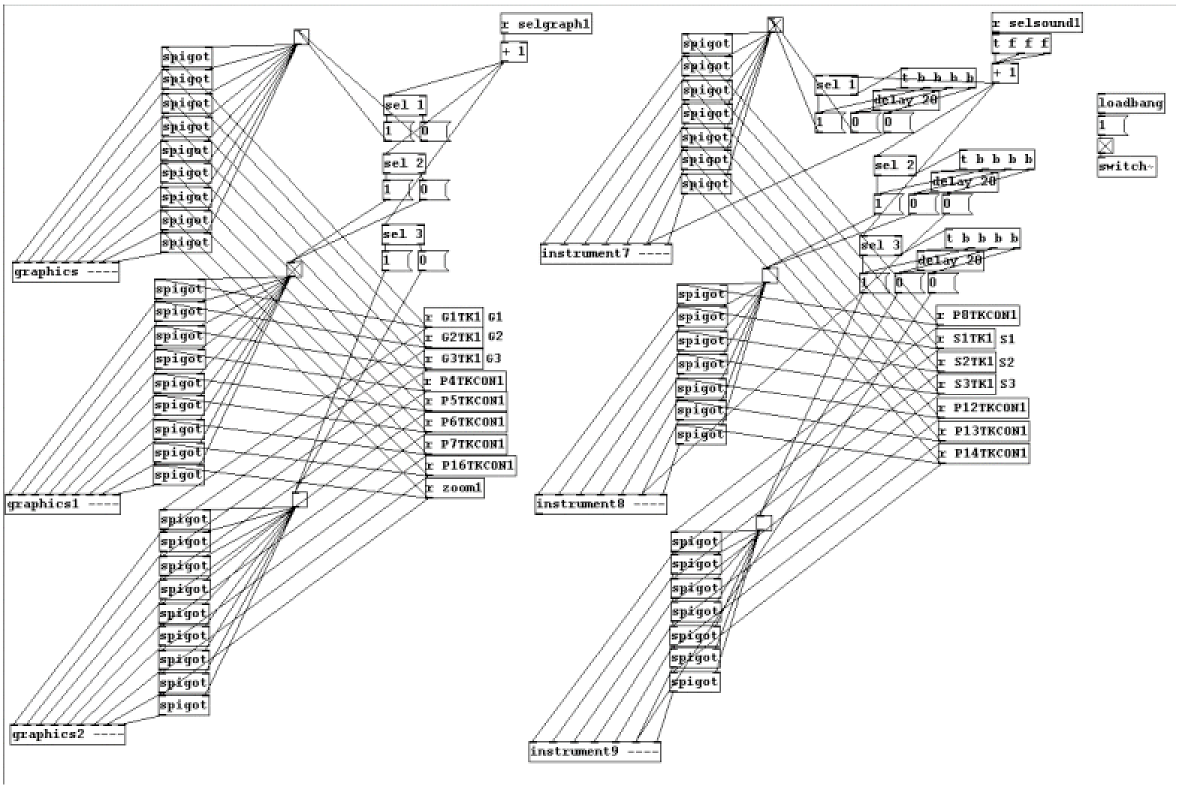


Figure 61: Graphics and Audio Synthesis switcher module.

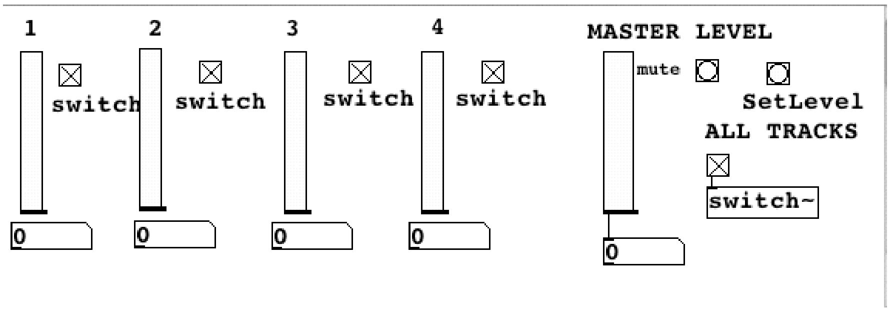


Figure 62: Master audio level module.

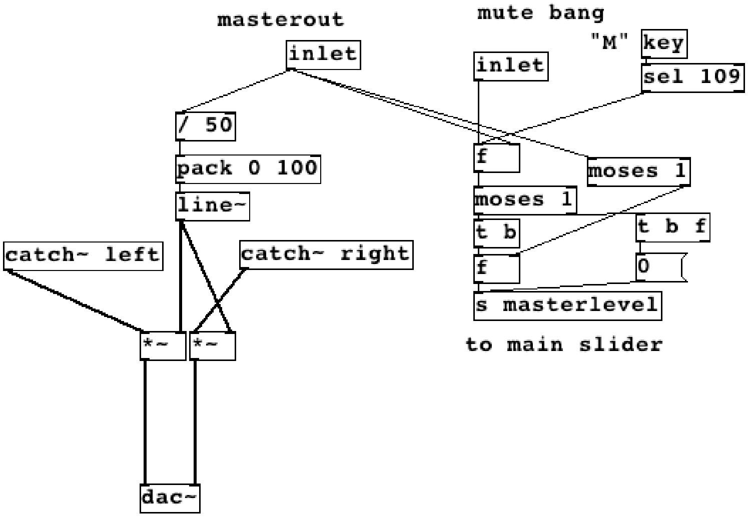


Figure 63: Audio Output module.

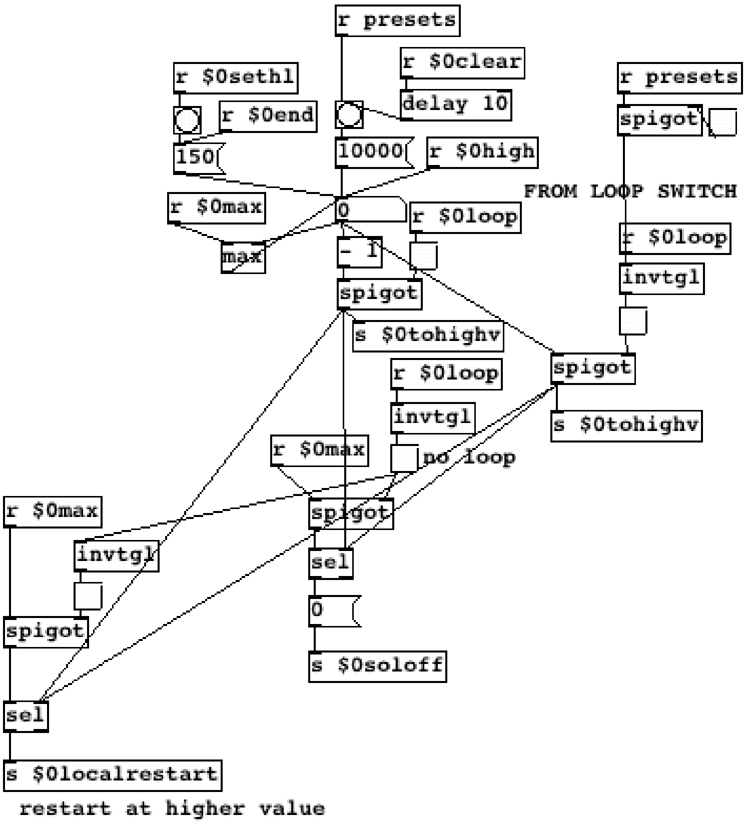


Figure 64: Loop module.

Timelines: Manages the visual representation of audio level, duration and starting point. Control of duration (stretch) and starting points.

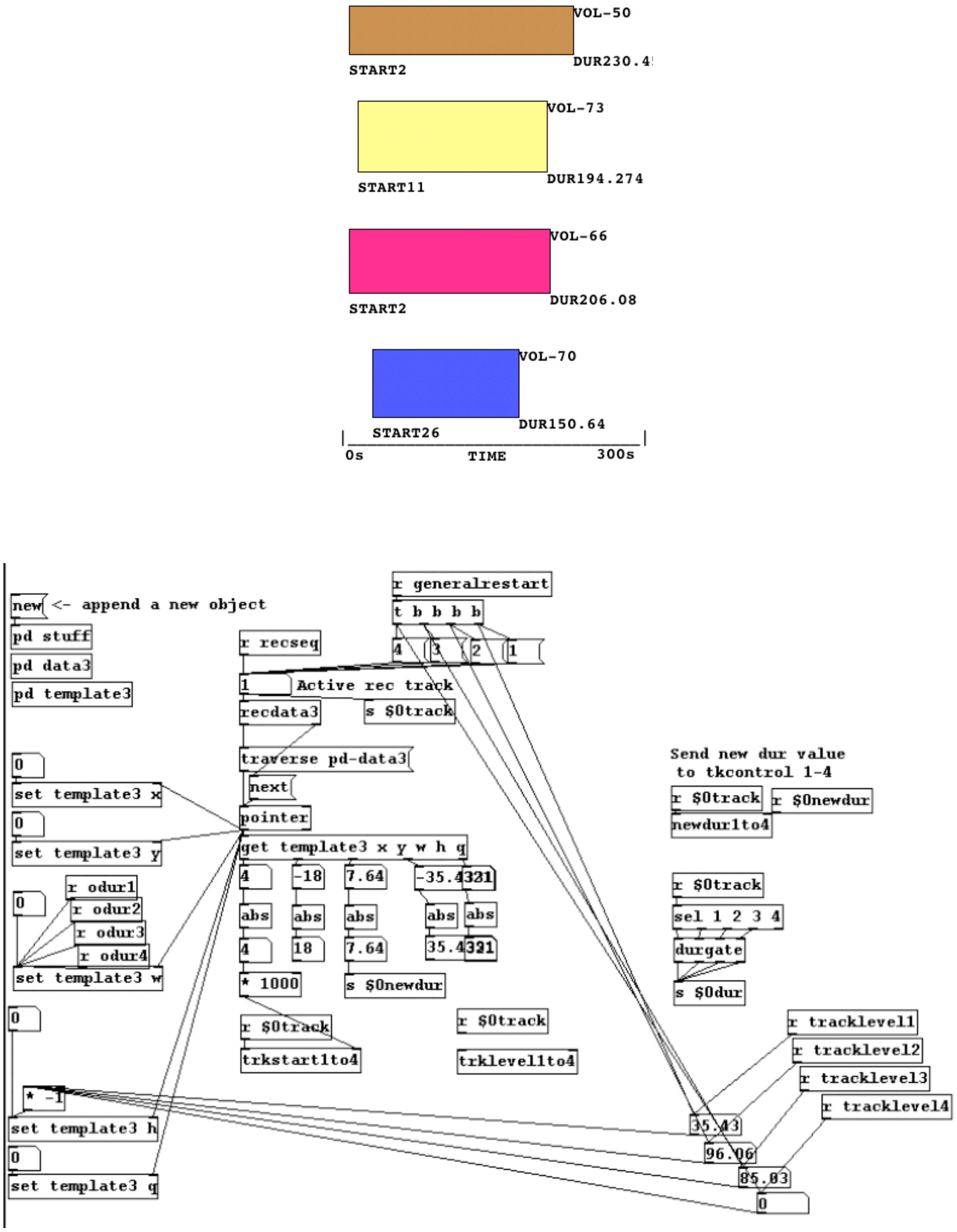
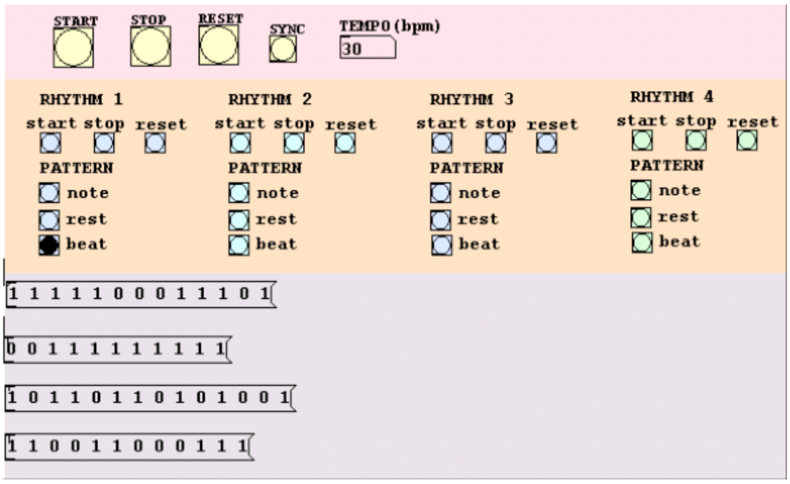


Figure 65: Timelines module.

Rhythm control: generate rhythm patterns.



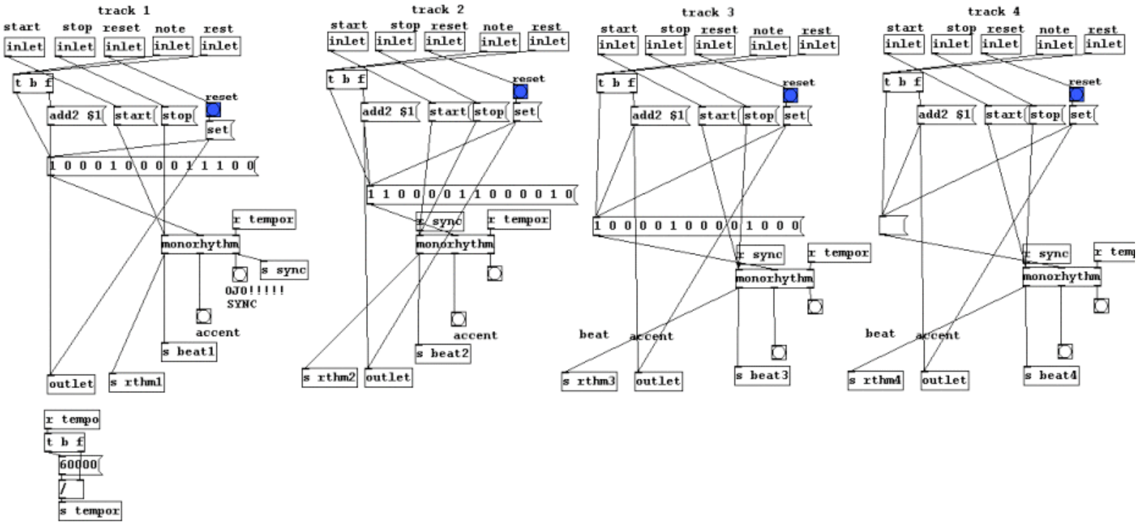


Figure 66: Rhythm patterns module. The core of the module is the *monorhythm* object.

Appendix E. CD-ROM containing the *Miró* application, demo movies and sounds.

The mappings between gesture - graphics, and gesture – sounds used in the demo movies are the same used in Appendix A.