DIGITAL THEREMINS: INTERACTIVE MUSICAL EXPERIENCES FOR AMATEURS USING ELECTRIC FIELD SENSING

BY

DAVID MICHAEL WAXMAN

A.B. Music
Univeristy of California at Berkeley
June 1990

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
In Partial Fulfillment for the degree of
MASTER OF SCIENCE IN MEDIA ARTS AND SCIENCES
at the
Massachusetts Institute of Technology
September 1995

Signature of Author____
Program in Media Arts and Sciences
9 August 1995

Certified By____
Tod Machover
Associate Professor of Music and Media
Program in Media Arts and Sciences

Accepted By____
Stephen A. Benton
Chairperson Departmental Committee on Graduate Students
Program in Media Arts and Sciences

Digital Theremins: Interactive Musical Experiences for Amateurs Using Electric Field Sensing

BY

David Michael Waxman

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on 11 August 1995
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

Abstract

Interactive computer music systems can allow amateur users to participate in facets of the musical experiences that would otherwise be inaccessible to them, such as performing, improvising, and playing with others. In this thesis, we state why we think this new access to music is worth persuit, and begin to explore how to make such interactive experiences sucessful. Eight example projects, called Digital Theremins because of their common electric field sensing user input, serve as a proving ground for ideas about interactivity for amateurs as well as a concrete framework in which to discuss them. Video examples of the projects and a diskette of the software accompany this document to aid the reader.

Thesis Supervisor: Tod Machover

Title: Associate Professor of Music and Media

Digital Theremins: Interactive Musical Experiences for Amateurs Using Electric Field Sensing

BY

David Michael Waxman

The following people served as readers for this thesis:

Reader____
Neil Gershenfeld
Assistant Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Reader____
Michael Hawley
Assistant Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Table of Contents

List of Figures and Tables

Acknowledgments

I would like to thank,

Tod Machover. His artistic talent, critical thinking, and unwillingness to take anything for granted has been a constant source of inspiration. Most importantly, he has been a good mentor, giving excellent advice, treating me with respect, and providing me with resources and encouragement to pursue my work.

Eric Métois and Josh Smith, who have not only become two of my best friends, but have taught me more about computers, math, and science than I ever thought I would know.

Neil Gershenfeld, Joe Paradiso, and Tom Zimmerman for creating the fundamental technology that made this work possible, teaching me how to use it, and putting up with me in their laboratory.

Mike Hawley, for supporting my work, reading my thesis, and filling the lab with fantastic late-night music.

Marvin Minsky, Mitch Resnick, who have provided some of the most interesting coffee machine chat imaginable.

Suzanne McDermott, Linda Peterson, Santina Tonelli, and Betty Lou Mclanahan.

All of the Hyperinstrument Group graduate students and UROPS.

My other mentors: David Wessel, George Lewis, Jean-Baptiste Barrière, Miller Puckett, and Cort Lippe.

And most of all, my family.

This Thesis is dedicated to the memory of Ruth Catherine Waxman

Introduction

Motivations

We normally consider the term "musical experience" to encompass a collection of separate, sequential steps: a composer creates a piece of music; he then communicates it to a performer in the form of a score or dictation; the performer in turn makes his interpretation of the composer's intent audible, either in concert or as a recording; then, one would hope, a listener hears the music and absorbs it with both his emotion and intellect. In *The Musical Experience of Composer, Performer, Listener*, Roger Sessions suggests that these distinctions have historically developed as "successive stages of specialization," and that passive listening is a relatively new, "even artificial means" of relating to music. In the past, Sessions goes on to say, composers and listeners did not think of music in terms of it being heard as much as in terms of it being performed.

> Even well into the nineteenth century the musical public consisted largely of people whose primary contact with music was through playing or singing in the privacy of their own homes. For them, concerts were in a certain sense occasional rituals which they attended as adepts, and they were better equipped as listeners because of their experience in participating, however humbly and however inadequately, in the actual process of musical production...[Sessions 50]

Today, we expect that a listener might be naive or untrained. Indeed, most of us neither compose, nor play an instrument, nor even dance to music; we simply hear it, paying as much attention to it as external circumstances permit. Those who do still engage in music making—be it playing the violin professionally, or getting together with friends to 'jam'—know the value of what the others are missing: experiences that are emotionally, intellectually, and socially gratifying in the most profound sense.

Moreover, as Sessions notes, making music does not only make one better at playing or composing, it enhances one's experience as a listener. An "understanding" listener does not just hear music, he also experiences the thoughts and emotions of composers and performers through music. He will, for example, imagine a cadence before it arrives. If a composer diverts the music to a different tonal center than was expected, or if a performer handles the cadence with particular elegance, this listener can appreciate the fact alongside them. He is, in some sense, their collaborator.

Unfortunately, to collaborate with one's ears alone is a difficult ideal, and it is infeasible to expect everyone to become proficient at playing an instrument or writing songs. Since Session's lecture at Juilliard in 1949, however, computers have become increasingly present in the "musical experience," and recently have enabled designers to create interactive music systems for novices. These are the subject of this thesis and, as we will see below, are precisely concerned with engaging listeners in a new and active relationship with music that is more accessible, more varied, and hopefully, richer than passive listening.

Digital Theremins

The Simplified Problem

Unlike professionals, who demand precise low-level control and are willing to spend the time to learn to wield it, novices must be given only a pertinent subset of that control that can be mastered in a relatively short period of time. To do this we must 'abstract' higher-level musical elements— things like tempo, embellishment, harmonic progression, or rhythmic activity—for real-time control. The requirements of this are twofold: figuring out which abstractions are appropriate for control by the user, and deciding the right way to 'connect' those abstractions to the interface so that they can be manipulated.

This Thesis

The Digital Theremins project consists of eight interactive experiments for amateur users which use gesture sensing for their interface. Each addresses a different aspect of the general problem described above, confronting both specific design challenges and broader questions about interactive music systems for amateurs. The ensemble of experiments, I believe, maps out a terrain of different possible approaches to the common goal of focusing a user's attention on a an experience that is 'musical.' One experiment, for example, explores the use of visual feedback, revealing a way in which it might be used to help an amateur grasp a universe of musical possibilities and improvise within it. Two others address the issue of ensemble performance, attempting to recreate the unique communication that exists between two musicians playing together. Another employs the idea of a gaming, combining musical and extramusical rules to challenge the user with a precise task and rewarding him with an interesting saxophone solo. A last experience gives the user refined control over many independent voices at once, each of which must be directed expressively to create a coherent whole.

Novel Interfaces

It is ironic that the Theremin, developed in 1920 and considered to be the first 'electronic instrument,' was actually more adventurous and comprehensive than most instruments developed since: it involved both a novel way to produce sounds (i.e., by non-mechanical means), and a radically new interface for controlling them. Since then, our ability to create, process, and manipulate sound has developed dramatically, while tools to 'perform' electronic music have progressed at a relatively slow pace. Today's commercial interfaces are most often fashioned in the likeness of traditional instruments. This model allows musicians to apply their technical skills to radically different sound production devices, but often creates an awkward and limiting relationship to the musical material actually being played. For example, the MIDI controller keyboard is excellent for playing chromatic pitches and harmonies; like the piano it mimics, however, the interface lacks expressive control over notes beyond their attack and duration, and is therefore completely unsatisfactory for playing non-percussive sounds.

In systems for novices, the physical interface is the user's window into the musical world presented to them. For these, we must be even more attentive to designing interfaces that posses a refined and intuitive link with the sounds that are produced. A keyboard key which plays a giant electronic gong simply does not feel as natural as an instrument which requires a large swing of the arm to make the same sound; and whereas a professional might be willing to 'suspend disbelief' in order to exploit and reuse his technical expertise, there is no such motivation for the novice.

Gesture and Music

In his book, *A Commonsense View of All Music*, John Blacking argues that movement is a basic means by which we express both what we feel and who we are, and that this expression is biological and involuntary:

> My Movements can express elements of the self that I have acquired through experience in society, such as tendencies to aggression, submission, domination, sensuality, or even a period of Military training. My movements can complement the feelings and movements of others who are present with me, or they can relate to the rhythms and sensations of the natural environment. [Blacking 87]

He goes on to assert that this expression through movement, especially the expression of excitement and ecstasy, is at the very core of the origins and purpose of music:

> The kinds of movement to which I refer may be called proto-dance. They are often accompanied by sounds, which I shall call proto-music. I may clap or slap my body, beat my feet on the ground or produce other kinds of noise by encountering some object in the course of a foot or hand movement...I am suggesting that dance and music are cultural developments of proto-dance and proto-music, and that one important purpose of these arts is to restore, if only temporarily, the open state of cosmic consciousness that is the source of their existence. [Blacking 87]

Whether or not one is willing to follow Mr. Blacking to his ultimate conclusions, it is clear that there is an intimate relationship between movement, music, and human expression that is more profound than would be suggested by any acquired or culturally specific training. We see this in the seemingly universal impulse to tap one's foot—no matter how 'unmusical' one claims to be—to a strong and regular beat.

Furthermore, people clearly express intentional and conscious emotion with gestures, both physical and vocal [Clynes 77]. If something is too loud, fast, strong, or agitated, for example, most of us know to hold our hands palms-down and move them slowly. Conversely, if something demands encouragement, vitality, or excitement, we hold our hands up, straight or in fists, and shake them vigorously. When language escapes us, we often turn to gesture.

Non-contact Sensing

Because of this apparent 'universality', we have chosen gesture as a constant means of input to all of the systems created for this thesis. Specifically, the projects take their input by detecting hand and body movements made in proximity to a non-contact electric field sensing device called the "Fish." This technology, developed by Professor Neil Gershenfeld and his colleagues at the MIT Media Laboratory, has several obvious merits (we will discuss the Fish in greater detail below in the *Projects* chapter):

- It is flexible in size, resolution, and number of input signals: this allows us to experiment with many mappings, configurations, and physical instrument designs.
- It is digital: the sensors output body position in a numeric form that is easily manipulable using a computer.
- It is non-intrusive: the sensors work without wires or other apparatus attached to the user.
- The inventors of the sensors are available and interested in musical applications of their work. This allows us to make modifications, experiments, and in general, to have an amount of support that is extremely beneficial for a project at this stage of development.

The Process

The original idea for this thesis was to write a single short composition, and 'test' several interactive system designs by using the piece as their musical material. In this way, I would not only be able to judge the merits of each individual design, but would create an interesting set of "interactive variations" along the way [Waxman 94b]. The problem with this approach is that interaction with music can take many extremely diverse forms. A system that allows users to trigger samples arbitrarily by making movements at discreet locations is a very different experience, in *kind*, than one that allows them to conduct a fixed score. It becomes difficult, then, to specify a "theme" that is both meaningful as a musical constant, and flexible enough to sustain different interactive approaches. This difficulty is compounded by the fact that the 'space' of possible interactions and their consequences is not yet familiar enough to clearly define a musical problem. In other words, composing such a piece requires that one knows at least a few rules and conventions—such as substituting harmonies or augmenting rhythms in a traditional theme and variations—so that they can be bent and stretched creatively.

I thus began the project by creating "experiments" to shed light on the interactive possibilities at my disposal. I soon realized that this space is potentially very large and mostly uncharted. Issues that seemed like 'implementation details'—such as finding a way to give users control over musical structure—exposed themselves over time as mammoth unanswered questions.

What began as preliminary 'tests' became, in effect, the research for this thesis: a series of experiments that begin to chart the space of musical interaction for amateurs. It is not as satisfying as the neat packaged "theme and interactions" might have been. I believe, however, that this preliminary experimentation was necessary, and that these subsequent pages outline many useful lessons, questions that had yet to be identified as important, and even a few answers that will aid in the future development of rich musical experiences for those who do not yet have access to them.

Background

Section Summary. There are two distinct sources of inspiration for this work: interactive music systems for performance by professionals; and a variety of forms of music for amateurs, both with and without computers. The former category has provided the majority of the technology and design ideas for these systems, while the latter has provided much of the inspiration and, along with observing users, has been the yardstick by which systems are measured.

Professional Systems

Electronic Instruments

The Theremin

The Theremin, developed in 1920[1] by Leon Theremin, is an electronic instrument that allows its performer to control a monophonic tone by moving his hands in the air [Galyev 91]. The Theremin's trademark sound is produced by the audible beating between two high-frequency oscillators (100kHz - 1 MHz). One oscillator is fixed, while the other is variable, its frequency controlled by the capacitive coupling of a performer's hand near an antenna. As the performer changes the distance between his hand and the antenna, he modulates the frequency of the variable oscillator, thus producing beats which create an audible pitch that changes with the difference between the two frequencies. A second antenna controls the amplitude of tone. The result is a thin whistling timbre whose pitch can be controlled quite precisely within the range of about four octaves. [Paradiso 95]

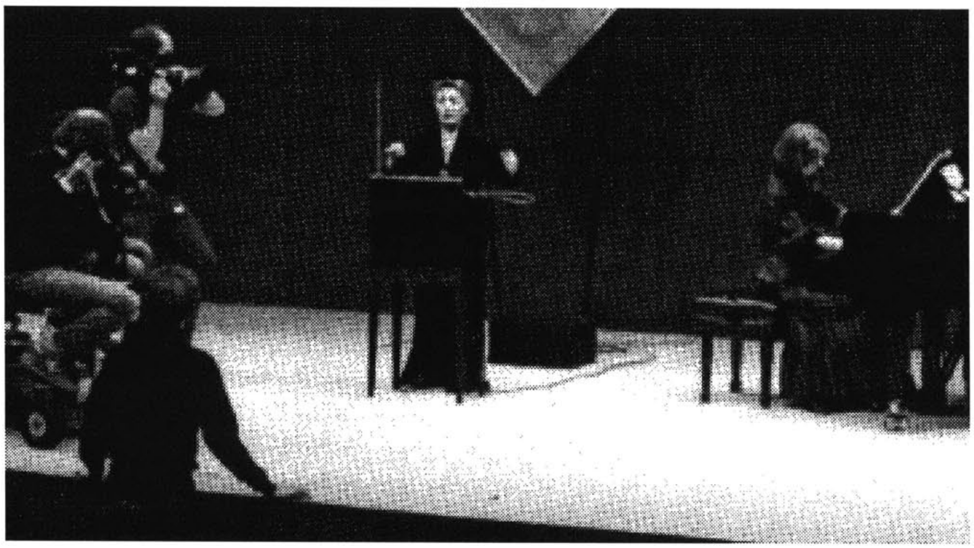
An Instrument Model

The Theremin, aside from its electronic circuitry, has much in common with traditional musical instruments. As opposed to the controller/generator paradigm of today's MIDI systems, the instrument's interface and the physical production of sound are both integral parts of a single system; and, like a traditional instrument, the Theremin must be mastered with patient practice. A few performers, most notably Clara Rockmore, a professional violin-

1. New Theremins are still being produced today by instrument/synthesizer designer Bob Moog.

ist by training, have become expert at the Theremin, and have used it to perform classical repertoire from Wagner to Ravel (as well as some new pieces created for it by composers such as Percy Grainger and Edgar Varese) at some of the worlds most prestigious concert venues.[Galeyev 91]

FIGURE 1. Clara Rockmore performing the Theremin at New York's Carnegie Hall.



With the exception of the Theremin and the Ondes Martenot (employed in some of Olivier Messiaen's compositions), most monotimbral electronic instruments invented in this century have failed to gain a place in the popular instrumentarium (and even these two are considered marginal). The problem is not only one of quality—these instruments still lack the richness of sound and control that traditional acoustic instruments posses—but one of critical mass: few performers are willing to put in the time to master an instrument that requires a totally new technique to play and has a near non-existent repertoire, and fewer composers are willing to write pieces for instruments that have no performers to play them.

Analog Synthesizers

The emergence of analog synthesizers represents a profound shift in electronic music history from 'instruments', to tools for the invention and performance of new timbres. Such instruments, like the *RCA syn-*

thesizer at Columbia [Darter 84], allowed composers to create a myriad of diverse sounds by combining and filtering variously generated electronic signals. These devices are not necessarily tied to a particular interface (though many are controlled by electronic keyboards). Predecessors to the modern analog synthesizer appeared as early as 1945 (with the *Hanert Electrical Orchestra*), developed as serious and useful musical tools in the fifties, and became 'popular' in the sixties (with the inventions of Bob Moog and Don Buchla) [Darter 84].

Interactive Computer Music Systems

Using analog synthesizers at first, and more importantly, with the invention of digital synthesizers (which allow generation and control of sounds to be mediated by a computer), composers have been able to create 'interactive' musical works: those which permit a dynamic musical dialogue between at least one participant and a system for electronically producing and manipulating sound. In a sense, this development echoes the 'instrument model' of the Theremin in that sound production and control are inextricably linked, but interactive music systems can provide a much more complex relationship between performer and production of sounds than the simple one-to-one mapping that the Theremin offers.

Many of Interactive computer music systems are intended for stage performance; thus, the people who "interact" are professional musicians and technologists. Tod Machover's Hyperinstrument pieces [Machover 92], and works such as Pierre Boulez's *Repons* [Boulez 81] or Philipe Manoury's *Pluton* [Waxman90], are examples of these. The performers who participate in the interaction are familiar with the system and score; they are—or should be—good listeners and outstanding practitioners of their "interface" (either their own instruments, modified to communicate with the computer [Machover 92], or specially designed interfaces [Machover 94b]).

Classifications

In his book, *Interactive Music Systems*, Robert Rowe [Rowe 93] describes several continua upon which one can classify the types of participation that interactive music systems provide. These classifications apply both to expert or novice systems.

- ***Score-driven vs. performance driven***

  Score driven systems respond to user input according to a predetermined musical score. This implies that the user's role in the interaction is to determine how (and sometimes when) composed musical events will unfold. The interactive pieces developed at IRCAM in the eighties using the 4x synthesizer (like the Boulez and Manoury compositions mentioned above), are primarily score-driven. Though they sometimes offer 'free' sections—like the middle movement of *Pluton* wherein the pianist improvises by adding notes into a melody generator—events are for the most part very precisely organized and defined in a pre-written score. A completely performance driven system, on the other hand, is one which offers sonic response to input without any particular score in mind. The subject of interactive discourse turns to the user's own experience, talent, and to his particular exploration of the rules of a system. Robert Rowe's *Cypher*, wherein input from a performer is analyzed for features and mapped to a series of preprogramed (but not necessarily pre-organized) output functions, is an example of a performance driven system. [Rigopolus 94]

- ***Transformative, generative, or sequenced***

  This continuum focuses on the source of musical material. Transformative systems take existing material, such as input from the user or a musical fragment from a library. The subject of dialog becomes the manipulation and mutation of this material. The system, "Jeux IRCAM" [Barrière 92], which allows users to sing a musical phrase and then manipulate it, is an example of a transformative system. Generative systems construct music in real time using predetermined rules and seed materials. DBX [Rigopolus 94] is an example of a generative system (we will discuss this system in more detail below in the section entitled *Interactive Music Systems for Amateurs*). Sequence based systems have fully constructed music or musical fragments stored in memory before the experience begins.

Michael Wu, in his thesis *Responsive Sound Surfaces* [Wu94], adds the distinction between compositional, performance, and experiential systems. This continuum focuses more upon the context of the interactive experience than the actual musical material. In compositional systems, the user is primarily a designer. David Zicarelli's M software [Chadabe 91] is an algorithmic music generator/transformer that the user controls in real time by interacting with a set of choices presented "cockpit style" on a computer screen. It is a good example of a compositional experience. A performance experience shifts the focus from design to execution, presumably for other participants or spectators. Tod Machover's Hyperinstruments [Machover 92] are examples of performance systems. In these pieces, the primary reason for interaction is to extend the performer's expressive control over preexisting musical material. Experiential systems are those wherein the user's primary responsibility is to have an artistic or educational experience, often extramusical. Most "interactive installations" fit this last category. Laurie Anderson's Handphone Table (When You We're Hear), a musical installation in which two participants hear music by putting their elbows on a vibrating table while clasping their ears, is a good example of an experiential system. Other examples include: *Sound Forest*, an outdoor installation by Christopher Janney in which participants walk through a large sculpture garden, triggering environmental sounds as they pass in front of photoelectric cells; and *Graphite Ground*, a piece by Liz Phillips in which participants navigate an environment of sensitive stepping stones, affecting pitches, rhythms, and timbres as they do so [Wu94].

Traditional Music for Amateurs

Salons, Folk Tunes, and Summer Camp

Salons

As stated above, Western classical music was much more of a 'hands on' experience during the Nineteenth century than it has become today. As the following account illustrates, amateur performance was not only reserved for the elite (who are so often depicted in 19th century fiction as retiring to the parlor to sing songs after dinner), but for the middle class as well. Note that amateur performance was, at least to this author in 1814, considered to be a 'female endeavor':

> Among the various refinements of the present enlightened age, the Science of Music appears, in an eminent degree, to have attracted the attention, not only of the exalted and affluent, but to have insinuated itself into the social enjoyments of every rank in Society.
>
> In the Modern System of Female Education, this fascinating accomplishment is very generally considered, as an indispensable requisite; and the Daughters of Mechanics, even in humble stations, would

> fancy themselves extremely ill-treated, were they debarred the Indulgence of a piano-forte. [Burgh 1814]

Due, perhaps, to the many alternatives vying for people's free time (beginning with the advent of the player piano, followed by radio, television, and now video games), the number of people who play chamber music today is relatively small. In classical music especially, the majority of our society has become estranged from the process of 'making music,' considering it a pastime for the cultural elite and a small handful of professional performers.

Folk Music

The tradition with which I grew up, and thus one that probably had much influence on this work, is that of folk music. Though folk is a broad term which applies to many styles and ethnic musics that are quite difficult to perform (like Appalachian bluegrass and Irish Fiddle tunes), there are many forms which are 'reserved' for amateur performance. These include drinking songs, work songs, and religious songs. A very interesting instance of the latter—and one in which I have frequently participated—is called Shape Note singing, which employs a simplified notation of shapes on a staff that is easy for amateurs to learn. Originally devised to teach sacred music—a context in which it is still commonly found—it has become a form of secular social entertainment that is still quite popular today.

> ...from very early, groups met in "singings" apart from worship services. Some singings were informal meetings of small numbers from a single parish or town...largest of all, and least numerous, were annual "conventions" lasting two days (often Saturday and Sunday) or even longer, and attracting singers by the hundreds. Characteristic of shape-note singings, which persist to this day, are the disposition of the singers in the form of a hollow square; unaccompanied performance, with trebles and tenors often doubling each other's parts; and the rotation among various singers of the responsibility for choosing the work to sing next, setting its pitch and leading the group (usually a first time through singing the solemnization syllables, then a second time with the text). [Eskew 86]

Summer Camp

Some of the best musical experiences for amateurs can be found in school yards, summer camps, and after school programs for children. For many, the greatest intuitive understanding of counterpoint comes not from listening to Bach fugues, but from singing *Row-Row-Row Your Boat* and *Frère Jacques* with a large divided group and feeling the magic of independent melodic lines falling into place as a unified canon.

Surprisingly, not one of the major music dictionaries which I examined had a listing for 'music games'. Children's games—like 'patty-cake,' 'ring around the rosy,' and rope skipping songs—are excellent examples of active and engaging musical experiences for novices. They require listening, coordination, and collaboration to play. As we shall see below, certain principles from these games—like having to achieve a goal while making music, or paying attention to other players and reacting to them—prove to be very useful for the design of interactive systems that use computers.

Interactive Music Systems For Amateurs

The Radio Drum

The Radio Drum, designed in the eighties by computer music pioneer Max Mathews, is a conducting instrument for both musical amateurs and professional performers. The instrument consists of two radio transmitting batons (tuned to two distinct frequencies), and a flat table-top surface embedded with receivers. Baton movements over the surface and features such as position, velocity and beat gestures are detected. Mathews uses the device to control the tempi and phrasings of traditional classical works in the manner of a conductor (i.e., by beating time note by note). As for a conductor, fluid movements with the second baton can be used to control features like dynamics and articulation.

Mathews conceived the radio drum for both professional performance (including new works by composers such as Richard Boulanger) and amateurs. For the latter, Mathews posits a new kind of "active listening" that will be achieved by conducting through pieces with the radio drum instead of listening to recordings of professional performances [Boulanger 90]. Though I wholeheartedly agree with Dr. Mathews goals, and am in general a great admirer of his work, I believe that giving novice users the same controls over expressivity that one would give a professional conductor is the wrong approach. Designers must do more than make new interfaces: they must rethink the relationship that novices can have with music, finding some middle ground that requires less physical dexterity and musical training than playing individual notes on an instrument, while requiring more attention and listening than turning on a compact disk player.

Drum Boy

This system, created by Tod Machover's group, with Fumi Matsumoto, Joe Chung, and Alex Rigopolus, allows amateur users to construct and organize rhythmic patterns using their aesthetic judgement. Unlike the Radio Drum, Drum Boy allows users to enter rhythms manually, or to recall patterns from a prerecorded database. Once rhythms are chosen, they can be modified in real-time by users with easily understood adjective descriptions such as 'mechanical' and 'graceful.' When one of these transformations is requested by the user (by pressing specially marked keys on a MIDI keyboard), Drum-Boy analyses the pattern that it is currently playing and modifies the music to make it more like the requested adjective. One interesting lesson from Matsumoto's work is that many novice users can discern what it is they like, and are able to 'direct' a musical texture towards that goal, even if they cannot play the texture themselves [Matsumoto 93].

DBX II

Like Drum-Boy, DBX II allows novice users to 'direct' musical textures that are generated in real-time by a computer. The system, created by Hyperinstrument group students Alex Rigopolus, Eran Egozy, Damon Horowitz, and Tod Machover, analyzes music for basic features such as syncopation and scale-tones, and uses the analysis to create similar musical textures. Users, by manipulating joy-sticks or an electronic keyboard, are able to intervene in the generation of music from the 'parametric seed,' and thus 'steer' the music from its original course. For example, by pushing a joystick forward, a user might add cross-rhythms to a previously straight-swing blues; by moving the joystick to the right, he can add more chromatic scale tones; and by pushing the button under his thumb, he can add timbral accents. [Rigopolus 94]

New Commercial Products

CD ROMs

With the exception of a few toys like 'Simon' (a memory game that requires users to repeat ever-lengthening patterns of flashing colored lights and synthesized sounds), there have been few commercial products that offer interactive musical games or experiences to amateurs. Recently, the invention and popularization of CD ROMs, with their ability to hold large stores of high-quality audio, have given rise to a plethora of commercial music products for the amateur. The first of these to enter the main stream were music teaching and history CD Roms, created by musicologist Robert Winter and published by the Voyager company. These CD Roms allow users to listen to audio samples, view scores and read historical and musicological information about famous musical works such as Stravinsky's *Rite of Spring* and Beethoven's *Ninth Symphony*.

The ability to have multimedia documentation for important works is quite appealing. However, there is nothing very new to this kind of presentation. Like traditional "introduction to music" college courses, these products often present musical structure like identifying the parts of a sonata form as the 'important' information to be learned, and sugar the lesson with fanciful narratives that explain 'what's going on' in the music ("enter the angry oboes with their quacking rebuttal of the second violins"). In the classroom, the strength of these courses is usually the excitement generated by the professors who teach them. This excitement, in the cases that I have seen, is then harnessed to teach more difficult and tedious musicianship skills (like clapping rhythms or sight singing). In this respect, musicology CD Roms fall considerably short of the music courses they imitate. This genre has remained popular, however, and many titles by Voyager and others (like Microsoft and Sony) continue to be published.

Another genre of CD Roms to arrive on the scene are what I call 'ego CDs'. These are published by pop stars like Peter Gabriel (the first artist to do this), David Bowie, and the Artist Formerly Known As

Prince. Like the musicology CDs, they offer sound clips of the artist's music and 'making of' historical documentation. Many, like the Gabriel and Prince CD Roms, also allow users to play with the music by remixing it. These CD Roms seem more intent on exploiting the burgeoning 'Multimedia' market with famous names than creating experiences that are new and interesting. They almost invariably use cliché navigation metaphors, like the artist's studio (Gabriel) or an imaginary palace (Prince), thus focusing the experience on the cult of the performer rather than on the music or the creative process.

Some (less) popular artists have made a concerted effort to use the CD ROM medium for its potential as something new. These, like Morton Subotnick's "All my Hummingbirds Have Wings" combine text and image in an abstract relation to the musical content, to produce a holistically conceived experience that would not be possible in any other medium. (It is not surprising that Subotnick has been a pioneer of other medium-specific composition earlier in his career, including works written for vinyl records.)

Rock Rap and Roll

A very few CD Roms have attempted to achieve goals more like those of this thesis: to let user's get their hands on musical material and play with it. The best of these that I have seen is Paramount Production's *Rock Rap and Roll*. This CD Rom permits users to arrange short audio clips (from libraries of stereotypical musical styles) to form new pieces. As the newly assembled music is playing, the user can trigger samples of sounds that are appropriate to the style by playing the computer keyboard or clicking buttons on the screen. User's can also record their own samples (using the Macintosh microphone), and add them to the texture. Though it is quite simple and the music a bit trite, this CD allows for creative play in a well defined, yet flexible framework. Users can improve the quality of their performances and 'compositions' over time and, especially because of the inclusion of recordable voice samples, can feel a sense of ownership about the output of the system. Lastly, the musical 'building blocks' that are offered to the user—i.e., song fragments and sample punctuations—are technologically appropriate for the CD ROM medium.

The Virtual Guitar

One of the several problems with the CD Rom distribution platform is that conventional computer interfaces are poorly suited to real-time musical interaction. My colleagues and I were quite excited, then, to hear that Virtual Music, Co. was releasing a musical game with its own specialized interface. When the 'Virtual Guitar' arrived, however, our enthusiasm was soon quelled. The instrument, a purple plastic guitar, is not much more than a traditional mouse controller, with a few buttons and some nylon strings that are about four inches long and make popping sounds when strummed. Users are asked to 'play along' with a famous rock performer by strumming the guitar to the music at times indicated by a graph on the screen. The game then rates users by giving 'points' for correctly played notes (i.e., those that are strummed at times corresponding to the graph).

Unfortunately, there is no way, other than learning by rote, to aurally predict when one is supposed to play. The visual graph, rather than helping user's match the rhythm of the music, is devoid of any useful information—like beat groupings or even a scale for correlating distance to time—and thus is harder to watch than to ignore. I witnessed several fine musicians 'score' very poorly on this system and be told by the famous rock star on the screen to 'keep practicing.' We can only hope that systems like this which discourage musical creativity and listening will not sour the general public to the idea of interactive musical games.

The Projects

Section Summary. The following section first describes the tools, both hardware and software, used for this project. Next, it details the eight Digital Theremin experiments in which the author participated as principle designer, and two projects that were realized by others with similar technology and design goals.

Toolbox

The 'Fish' Sensors

This thesis has largely been made possible by the invention of a new human-computer interface device by Professor Neil Gershenfeld and his colleagues in the MIT Media Laboratory Physics and Media Group. Like the Theremin, the interface device (dubbed 'Fish' for its similarity to the sensing behavior of weakly electric fish) uses a small electric field to detect the position of a body in an open physical space. The Fish are both inexpensive and extremely flexible:

> The sensors are low power (milliwatts), high resolution (millimeter), low cost (a few dollars per channel), have low latency (millisecond), high update rate (1kHz), high immunity to noise (>72dB), are not affected by clothing, surface texture or reflectivity, and can operate on length scales from microns to meters. [Zimmerman 95]

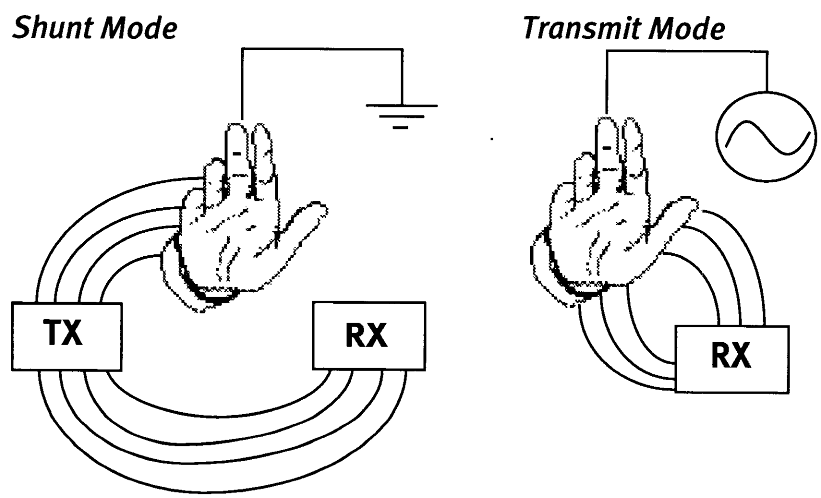
The Fish evaluation board is equipped with analog to digital converters and a microprocessor that translate sensor signals to easily manipulable MIDI control messages.

Modes of Operation

There are two distinct ways to use the Fish evaluation board: shunt mode and transmit mode. In shunt mode, an electric dipole field is created between two electrodes[1], one an oscillating transmitter and the other a receiver. When a user moves his hand (or anything else that conducts electricity and is reasonably grounded) between the electrodes, he shunts some of the signal to ground and thus decreases the amount of current detected by the receiver. This reduction in current is translated into a decreasing 7 bit MIDI controller value by the evaluation board's processor[1].

The sensors can also be used in 'transmit' mode. In this mode, a user is capacitively coupled with the Fish transmitter (in other words, he must touch it or be extremely close to it), and effectively becomes a transmitting electrode himself. As he approaches a receiving electrode with part of his body, the amount of current received (and the channel's corresponding MIDI signal) increases.

FIGURE 2. The two modes of Fish sensor operation.



Though the latter mode is easier to map because there are more direct relationships between the spacing between transmitter and receiver, there are advantages to leaving the user completely unencumbered and mounting electrodes solidly in a non-conducting structure. This is especially the case for public space exhibits or systems

1. Electrodes are simply metal plates or pieces of conductive foil that are connected by a shielded cable to the Fish evaluation board's transmitter or one of its four receive channels.
1. The evaluation board is also equipped to send out an 8-bit digital signal through its built-in serial port. This requires a different program to be loaded into the board's microprocessor.

which require a lot of movement to operate. For these reasons, we have mostly used shunt mode.

Normally, we employ the sensors to detect a user's hand position and movements within the space of a few cubic feet. In the most basic scenario, position in one dimension is derived from a single sensor's raw output. An array of these can be used in order to capture several degrees of freedom in different parts of the space. More often than not, custom software tools are used to convert the raw output from multiple sensors into more interesting and usable information. This can range from simply smoothing the raw data, to calculating position in three dimensions, to extracting other information about hand movements such as how jagged or rapid they are.

Software Tools in The Max Programming Language

All of the software created for this thesis was written by myself, with considerable aid in algorithm design from Eric Métois, Josh Smith, and Michael Wu. The software is written in Max [Puckette 90], a high-level graphic programming language specially geared towards applications involving real-time MIDI processing. Because of its built-in MIDI functions and rapid interface prototyping capabilities (such as easy window and button creation in an interpreted environment), it is an excellent tool for testing and development of interactive music systems for amateurs.

The Patch

A Max program, called a 'patch,' consists of a collection of instances of primitive objects connected together with 'patch chords' that represent data flows. The standard collection of primitives (called 'objects,' though Max lacks some of the features—like inheritance—that one normally associates with object-oriented languages) vary from low-level mathematical functions, to high-level self-contained programs such as sequencers and MIDI file parsers.

Custom Objects

Max objects can be created by the user—either in MAX itself or in C—and added to the programming environment as extentions. These custom objects can be told what messages to expect and how to handle them. For this project, several objects were created for use as general tools for working with sensors and music. These fall into two categories: sensor tools and musical tools.

Sensor Tools

The first category of tools deal with the input and processing of information from the sensors:

- sensors_in
- velocity
- beat
- acceleration
- in?
- gesture_rec_xmit
- active
- jagged
- low_pass

sensors_in

This object takes raw sensor data from the MIDI stream, polls it at a constant rate (normally 30ms), displays it in a window both as sliders and numerically (for calibration and testing), and sends the values out to the rest of the patch under the labels sensorA1, sensorA2, etc. These data are considered 'raw' position values in one dimension or sensor "zone."

velocity

This object is used to determine the rate of change in sensor values. It takes each value from sensors_in and subtracts the previous value from it. The result is normalized to a suitable range.

beat

This object detects 'beat' gestures in a sensor zone. Beats are determined by a simple velocity threshold (normally a negative velocity for shunt mode). In this way, beats can happen at any location in the sensed zone (as opposed to using position threshold like Max Mathews does for his Radio Drum). By fine-tuning the threshold, one can trigger beats with a slight anticipation of the point where one's wrist normally snaps, and thus compensate for any lag-time inherent in the system. To avoid multiple unwanted beat triggers, a two-step process by which beats are armed and then triggered is employed.

acceleration

This is determined by two velocity objects in sequence.

in?

Each sensor field has an 'in?' object with an independent threshold value. If a sensor value is below the threshold (or above for transmit mode), a flag is set to signal that a zone has been entered.

gesture_rec_xmit

This object contains the above objects with values inverted for use with transmit mode.

presence

Whenever a sensor field is entered (determined by 'in?'), the presence value for that field slowly increases towards a maximum (the rate of increase can be controlled by the designer). When the user leaves the field, the presence value slowly decreases to zero. This is useful for determining how much a user interacts with a given sensor zone.

active?

This object determines whether or not a sensor zone is 'active' (i.e., when the absolute value of a zone's velocity is more than a given threshold).

low_pass

This first-order Infinite Impulse Response low-pass filter object has been very useful in all circumstances where one wants to see general trends and ignore intermittent spikes in data (see the Jagged example below).

$$V_n = (1-\alpha)\,U_n + \alpha V_{n-1} \qquad \textbf{(EQ 1)}$$

jagged[1]

This object is used to determine the 'jaggedness' of a user's behaviour in a two-dimensional sensor space. Jaggedness is derived from the square of the magnitude of acceleration in two dimensions, filtered using the low_pass object defined above. If X_n and Y_n are, respectively, the positions on the x andy axis at time n, then we define:

1. The mathematics for the determination of Jagged was developed for the Bach-o-Matic example by Eric Métois.

$$\text{Acceleration on the x axis: } X''_n = (X_n - 2X_{n-1} + X_{n-2}) / \tau^2$$
$$\text{Acceleration on the y axis: } Y''_n = (Y_n - 2Y_{n-1} + Y_{n-2}) / \tau^2$$
$$\varphi_n = X''^2_n + Y''^2_n$$
$$\text{Jaggedness: } J_n = (1 - \alpha)\varphi_n + \alpha J_{n-1}$$
(EQ 2)

Musical Tools

Each project has led to the development of its own unique musical environment. Indeed, mapping musical responses to sensor input has represented the bulk of the programming. However, a few tools have proven general enough to be used in several projects.

- trans_scale
- ambi
- chord_play
- tempo_maker
- change_sections
- control_sequence
- convert_neume

trans_scale

This object takes incoming chromatic notes and quantizes them to the notes of a scale (i.e., each note is transposed to the nearest scale tone). Designers can define their own scales as Max tables. trans_scale can also be made to transpose notes (after they are mapped to the new scale) by passing a transposition value (in semi-tones) to the object's right inlet.

ambi

This object takes incoming notes and restricts them to a given range or ambitus. Notes which fall outside of the bounds are 'wrapped' back to the nearest note that is of the same pitch class and is within the upper and lower bounds defined by the designer.

chord_play

This object plays predefined chord progressions that are stored in a Max 'coll' object. Whenever chord_play receives a 'bang' message, the next chord in the progression is played.

tempo_maker

This object clocks the time between sucessive incoming beats and outputs their tempo. The result is low pass filtered (using the low_pass object) so that the tempo values change smoothly. A maximum tempo can be set manually that causes the object to ignore large pauses.

change_sections

This object performs simple sequential 'mode' changing from one mapping to the next. It takes a mode number (normally supplied by a counter object in the linearly organized pieces) at its input and takes care of the necessary on-off messages associated with that mode.

control_sequence

This customized sequencer takes a file of note events in 'list' form (i.e., notes are listed with durations as opposed to separate 'on' and 'off' messages) and plays them as a sequence. Unlike a traditional sequencer, individual voices have independent scaling of tempo, duration, transposition, and dynamics.

convert_neume

This object converts a small melodic fragment to a musical 'neume' a representation of melody in terms of interval contours as opposed to exact pitches. For example, the opening half-phrase of "Twinkle Twinkle" would be converted into 'no-step, medium-step-up, no-step, little-step-up, no-step, little-step-down.' The interpret_neume object then reconverts the contours into exact pitches by matching step sizes to intervals. This is done with weights: a big step might most often be interpreted as an octave, half the time as a major tenth, very rarely as a minor ninth, and so on. By adjusting these weights, one can produce interesting variations of a melodic fragment.

Prehistory: Hand Gesture Music

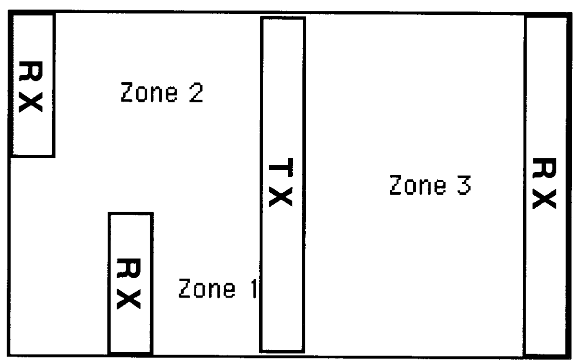
Hand Gesture Music was created in the Fall and Winter of 1993 in collaboration with graduate student Michael Wu. It was our first attempt to use the fish sensors as an interface for musical amateurs and, in many respects, was our initiation to the subtle and not so subtle difficulties posed by such an endeavour. The project was originally conceived as a system for novices. However, in our excitement to try new and complex mappings, we let the project evolve more as an interactive composition, requiring both practice and skill to perform. For the designers, Hand Gesture Music became a proving ground for interactive experiments, many of which led to the development of tools and thinking used in later, more refined projects. Because the project provided a foundation for so much of our later work with sensors, we will describe it in some detail.

The system

Interface Configuration

The sensor interface for the project consists of a simple, flat plexiglass surface embedded with one transmitter electrode and three receivers of varying sizes and placements, creating three active sensor "zones", each with an independent measurement of position, velocity, beats, and presence.

FIGURE 3. ***Hand Gesture Music*: Sensor Layout.**

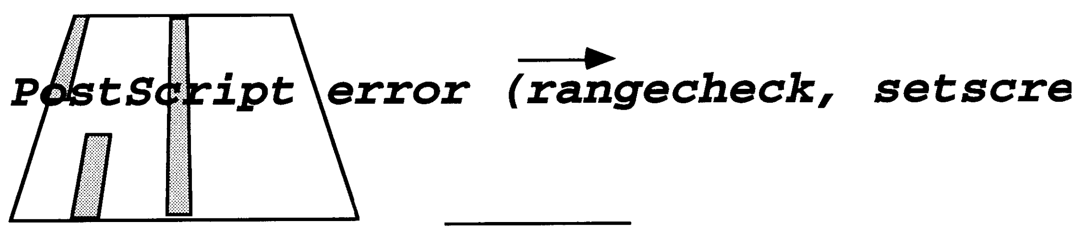


Computer Systems

In its original form, the project required two Macintosh computers: one dedicated entirely to sensor input and processing (using an add-on A to D card) and a another used to make high level musical decisions and control the sound hardware. The 'sensor input' computer, with software written in *Smalltalk* by Michael Wu [Wu 94], takes raw data from the sensors, smooths it, extracts velocity, beats, and presence, and sends the "polished" results as MIDI controller information to a second Macintosh which makes musical decisions (using software written in Max) and sends MIDI messages to the synthesizers and processors. In later projects (and a revised version of this one), the development of the MIDI fish allowed us to use only one computer for sensor input, musical processing, and sound control.

One of the MIDI playback devices is in fact a general purpose computer, an SGI Indigo running a real-time sampling program written specially for the project by Eric Métois. This sampler enables 'on-the-fly' sampling, trimming, and playback of sounds input to a microphone that is also part of the *Hand Gesture Music* system.

FIGURE 4. *Hand Gesture Music:* Diagram



The piece

Structure

The piece is organized as a sequential progression of three major sections, each with a collection of subsections and interactive 'engines' that map user input to musical responses. The first section, consisting mainly of sampled metallic percussion sounds such as bowed cymbals, gongs, and Gamelan instruments, has two primary modes of user interaction: *beat triggers* and *beat texture building*.

Beat Triggers

Beat triggers provide an extremely simple, yet effective form of interaction that is clear to both performer and observer. Beats detected in one of the three sensor zones are used to trigger a single sound (like a gong strike or cymbal) or musical gesture (such as an arpeggio or tri-chord).

Beat Textures

At the same time, the largest sensor zone controls a beat texture engine that generates rhythmic background for the section. Like the triggers, a single beat in this zone creates a single percussive sound (a sampled Gamelan strike in the mid-range). The pulse entered by the user is then quantized to a twenty-four beat looping cycle where it remains. In this way, users can add beats to a continuous loop and build a texture of accent patterns.

Moving Along

There are four such layers of beat texture, each independent and sounding on a unique pitch. The problem in designing this part of the system (and it has been a problem on many occasions since), was how to give the user control of which layer he affects. More generally, this addresses the question of allowing the user to control 'structure' as well as 'texture.' Our simple answer to this larger issue was to organize the piece linearly, thus reducing the problem to finding an unmistakable gesture to trigger an advancement from one micro-section to the next. To do this, we use a vocal trigger which will be explained below.

SGI sampling and Volume Control

As stated earlier in this section, an SGI computer is employed as a real-time sampler using custom software written by Eric Métois. This sampler is also able to return a MIDI value corresponding to the amplitude of input to the microphone. This feature is used to detect when the user has entered a sample and, when the sample is recorded, advance to the next section of the piece. Thus, after building one beat texture layer satisfactorily, the user enters a sample and begins working on a second layer. This process continues until all four layers are constructed and playing simultaneously. Subsequent sample triggers further advance the piece, by changing the pitch set of the layers, adding instruments, and so on.

Voice Manipulation

The thinking behind using samples to advance the piece was that we could "collect" them for future use as musical material. After the buildup of added rhythmic layers at the end of the first section, the texture calms and the samples return as the primary musical material.

Continuous Control

As opposed to the first section, wherein control is based on beats, and the music and motion are jerky and angular, the second section is fluid and calm, and uses continuous sweeping gestures for control. Vocal samples collected from the first section are played back automatically, and indexed by the position of the user's left hand. If the left hand is close to the surface, only the first few samples are played. As the user moves further from the surface, he 'opens the space of samples' to include the entire recorded collection. These samples are then played back through a commercial signal processor (a Lexicon LXP-15) which adds a glide delay that is controlled by the position of the right hand. In moving in and out of this controlling zone, the user can change the period of the reverb to make a flange effect, or 'feel out' resonances (the fast constant delay behaves like a band-pass filter, and the center frequency can be tuned to create 'ringing'). Lastly, the presence parameter is used to control the wet-dry mix of the processor. The longer the user manipulates the samples, the more strange and distorted they become. To return the samples to their original form, he must pull his hand completely out of the field and let the sound 'calm down' slowly.

Weighted Pedal Tones

The third section has many elements (both interactive and musical) similar to the first, but it is more tonal. This tonality is underscored by a pedal tone that is sustained by a sampled 'cello sound in octaves. To give the pedal tone color and richness, partials of the sound are accentuated and brought out with a mix of flute and string sounds, and the position of the hand over the whole table is used to determine which partials are to be accented. In this way, the user can modulate the timbral quality of the pedal tone while performing the rest of the triggers and continuous sounds played in the section.

Collaboration: The Gesture Cube

Even before we began *Hand Gesture Music*, we imagined an instrument/experience for more than one player. We wanted to give novice users a taste of the exceptional and unique experience of ensemble performance and improvisation. The *Gesture Cube*, built in collaboration with students from Tod Machover's "Projects in Music and Media" course and Sharon Daniel's "Video Art" course[1], was an attempt to create such an instrument.

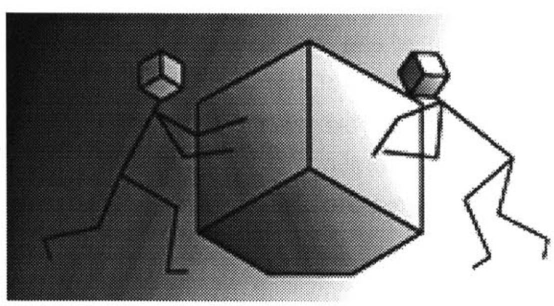
Public Space Context

Unlike *Hand Gesture Music*, which remains to this day a demo that is too complex for amateurs to perform, the *Gesture Cube* was truly intended as a 'public' piece, presented as part of a larger installation of interactive music works organized by Professor Machover and presented in the MIT Media Lab for three days as *Voice Motion Experience*[2].

Musical and Architectural Material

The project is based on the three themes of the exposition: the human voice (from which nearly all of the sampled sounds were produced), motion (the mode of interaction with the instrument), and the shape of a cube (the show took place in the Experimental Media Facility at MIT, affectionately known as 'the cube' because of its near-cubic dimensions). The instrument is constructed as a 4" x 4" cube made of canvas and wood, and turned upon one corner that seems to be 'lopped off'. Two of its faces are wired with one transmitter and two receivers each, while the third is used for video projection from inside the object.

FIGURE 5. The Gesture Cube



Music and Interactions

There were only a few significant advancements in interaction for the cube project. It depended largely on beat triggers and continuous

1. The team consisted of Maribeth Back, Jin Kang, Adam Lindsay, Joshua Smith, and myself.
2. AKA, Vox[3], May, 1994.

sound modifications using simplified tools developed for *Hand Gesture Music.*

In an attempt to clarify the two performers' independent roles, sounds were partitioned into two categories, percussive and continuous, and each category was assigned to a different playing face. Unlike *Hand Gesture Music,* wherein sections dovetail into one another with the addition and subtraction of musical elements, the cube is organized–again for simplicity–into six static, self-contained sections. All of the even numbered sections are quite similar, while the odd sections are more varied, thus forming a simple looping *Rondo* form (A B A' C A'' D A B, etc.).

Changing Sections: a Gesture of Collaboration

Once again, our biggest problem in offering such a system to the public was giving over control of structure, and once again, we chose a linear form with a clearly definable gesture to move from one section to the next. Despite our inclination to make the piece be self-explanatory, we decided to make the 'section-advance' gesture a very deliberate action that had to be taught by *Vox-cubed* attendants[1]: both players put both hands completely in the field—at which time a previously unheard Tibetan bell sound signals them to step completely out of the field. When they reenter, the next section begins.

Though this gesture was initially conceived to avoid accidental section changes (and because it is a clearer solution for the players than some statistical methods of changing sections that we considered), performing the obligatory communal gesture turned out to be one of the most satisfying parts of the experience.

Video Control

The *Gesture Cube* was also our first attempt to correlate image with music and gesture. A video made for the project by Jin Kang and Joshua Smith is projected on the third (unplayable) face of the cube. Smith describes the content of the video as follows:

> The video is a meditation on the subject of hands as instruments of expression. It is a sampler of manufactured—literally, made-by-hand–

1. In fact, when traffic in the space was high, people generally learned this gesture by watching other users.

–products: linguistic gestures, woven cloth, the Gesture Cube itself, and the musical sounds that the participants have produced, by hand as it were.[1]

This video is broken into sections that correlate to the sections of the piece. When users change sections, the video (recorded onto a video disk) changes with them.

Design Flaws

Though the *Gesture Cube* received many compliments and, in general, seemed to be enjoyable to play (indeed, the enthusiasm generated by the project has led it to be installed in several locations since *VOX-cubed*), there are many aspects of its design that defeat its purpose of promoting ensemble communication between two players. In terms of lessons learned, it is certainly the most important predecessor to the *Frames* project described in the end of this section.

An Ill Suited Physical Interface

Despite the fact that players can see each other's faces, they do not have a view of each other's hands and bodies. This corporal communication turns out to be very important, and its absence leads to confusion about "who is doing what" to produce sounds. Further, neither player can see the video projected on the third face, so they cannot appreciate that they are controlling it. (In fact, many onlookers try to "play" the inactive video face.)

No Correlation of Gestures

The attempt at constructing an instrument for two people was limited to an exercise of sound design. In other words, we thought that making two instruments that would naturally sound 'good' together was enough to encourage listeners to pay attention one another and play as if the cube were a unified ensemble. More often than not, unfortunately, players concentrate on their part with no feeling of trying to make a musical whole out of the experience.

The Flailing Arm Syndrome

Because the two instruments make a lot of complex sounds, and because, for reasons described above, each user has difficulty in identifying which sounds he is directly producing, users do not seem

1. Joshua Smith, personal communication.

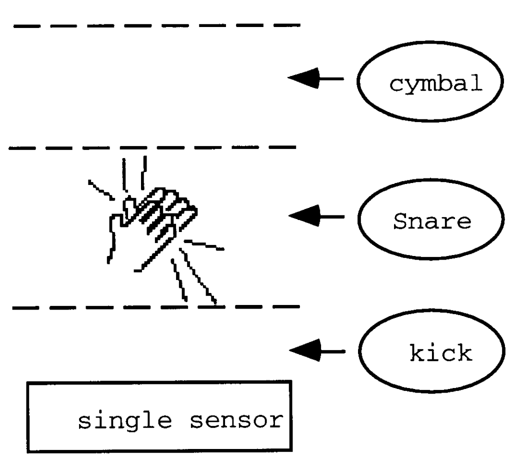
to have a good sense that they actually control the music that emanates from the gesture cube. Participants do realize that "waving their arms" causes music to output, and a few discover and learn some of the more apparent mappings, but it is very hard to progress beyond that and actually 'play' the instrument with musical intention.

Combining Inputs: The Clappatron

The clappatron is a small-scale intermediary project intended to pursue the idea of combining audio and sensor input to a system. In this experiment, I once again employ Eric Métois' sampler program, but only for its amplitude output. Using a modified version of the beat tool, spikes (corresponding to claps) in the audio input are detected and used to trigger percussion sounds. The timbre of the percussion sounds are chosen by dividing the position (raw) value for one sensor into four discreet vertical zones. By clapping above a surface at different heights, one can play one of the four drums with a considerable amount of control.

This experiment proved quite satisfactory, I believe due to the simplicity of the mapping and the tactile satisfaction of clapping.

FIGURE 6. The Clappatron: interface diagram



Sound Design or Music Game?: MPONG

The idea behind MPONG was to create a musical game by mapping significant interactive sound design to a proven video game: Pong. Fish sensors were used to control paddle positions, though this was not the primary focus of the exercise.

How it Works

The musical material generated by MPONG is a simple 12-bar blues: the Y position of the ball is mapped to a blues scale (with higher on the screen corresponding to higher in pitch); the background bass line and drums are automatically played whenever the ball is in motion; and when the ball hits a paddle, a harmonically correct chord is played.

Why it is Interesting

One could convincingly argue that MPONG is more of a game with music than a musical game. As opposed to the other projects described in this section, it is the only one that is still comprehensible with the volume turned down. What is remarkable, though, is that the music completely changes the object and amusement of the game. The goal of the players immediately shifts from getting the ball past the other player to maintaining a rally. Furthermore, because the musical response is more interesting when the ball takes a steep-angled trajectory, players are inclined to strive for complexity and difficulty of play.

An Hypothesis

A better Pong game program would have allowed me to sound design behaviors like ball spin and paddle velocity. I suspect that one could, in this way, give control of a significant number of fast moving musical parameters, and that the user would be able to handle them more easily than he might without the "anchor" of the Pong game.

Timbre Control: The AFKAP Frame

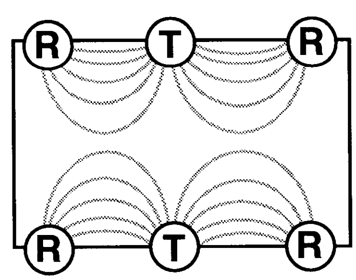
This project arose from a request by The Artist Formerly Known as Prince (hereafter referred to as AFKAP) for a sensor instrument to use in his stage show. He had originally asked our group for a copy of the Penn and Teller Sensor Chair (see below), but we chose to provide him with a more general system for experimentation. Though he is by no means an amateur musician, I wanted to make a simple example instrument that he could immediately play, understand, and modify for his own purposes.

The Instrument

The physical instrument, designed and built by Joe Paradiso, consists of a frame made of PVC tubing with six rings of copper pipe as electrodes. The two center electrodes were connected to a single transmitter, while the four corners served as receivers. This unusual doubling of the transmitters was not very helpful in the end, because the overlap between fields made it difficult to interpret hand position. It is not a configuration that should be repeated.

The idea behind the shape of the instrument was that AFKAP could peer through the frame and gaze upon his adoring audience, and that they, in turn, could see his hand movements. It turned out to be quite an appealing design, not only because of its aesthetic value as an object, but because it clearly defines the sensor space for both performer and audience. It is a design that we would use again in the *Sensor Frames* project described below.

FIGURE 7. The AFKAP frame



The Music: Timbre Control With the VL1 Synthesizer

Prior to this project, most of the sound generation and production for this research was achieved with samplers. This choice was made because of sampler's sound quality and the ease with which one can construct one's own timbres with them. Samplers, however, offer little real-time control apart from triggering notes, volume, and pitch bend. The desire to have users "get inside" timbre and manipulate it was thwarted by these limitations.

In this project, a radically new synthesizer (called the VL1 and produced by Yamaha) is used. The VL1 generates sounds according to physical models of instrument systems and affords an enormous amount of continuous control over sounds once they're launched. Moreover, since sounds are created using physical models, modifying instrument parameters in real-time can lead to unexpected, natural sounding results (such as reed-squeaks) and variation of timbre.

The idea of physical modeling synthesis has been around for some time. However, it is extremely computationaly expensive to implement, and thus, until recently, has only been available in deferred-time software packages running on powerful workstations. The difficulty in using physical modeling synthesis for musical projects is that it requires constant continuous control (a perfect violin model, for example, would require as much input from the user as a perfect violin). Without the feedback offered by a real-time system, this refinement of control is hardly possible.

With real-time feedback, controlling physical modeling synthesis is still quite difficult. Yamaha has alleviated this to some extent by consolidating control parameters into pertinent clusters that are intuitively accessible to musicians (such as 'tonguing' and 'growl'), but control is not optimal with a standard keyboard interface.

The sensors, on the other hand, can easily be used to generate several simultaneous streams of control information. Though it is difficult to generalize about effective mappings (much depends on the timbre and sensor configuration in question), simply keeping several controller values moving seems to produce rich and interesting results.

A One Sound Instrument (Almost)

Thus, the AFKAP frame project focuses mainly on the exploration and control of a single timbre: a rather raunchy electric guitar sound that offers several control parameters. In order to create a backdrop and harmonic support for the guitar, a two bar looping groove was sampled from one of AFKAP's recent albums.

The groove commences playing whenever a sensor field is entered. To start the guitar sound, the user has to provide a beat in one of the sensor fields. Once launched, the timbre and pitch are controlled by the position values of three of the sensors. This represents a nearly direct mapping to synthesizer parameters, with some modifications made to response curves, etc.

The fourth sensor zone is reserved for punctuations: expletives sampled from AFKAP's album that are launched with beats to the zone.

A Simple Effective Instrument

In comparison to other projects, the AFKAP frame provides a very simple direct mapping between movement and low-level timbre control. More like an instrument than an interactive composition, the frame proves quite fun to play and encourages careful listening on the part of the performer to modify the instrument's behaviour.

Visual Aids: Barbie.pat

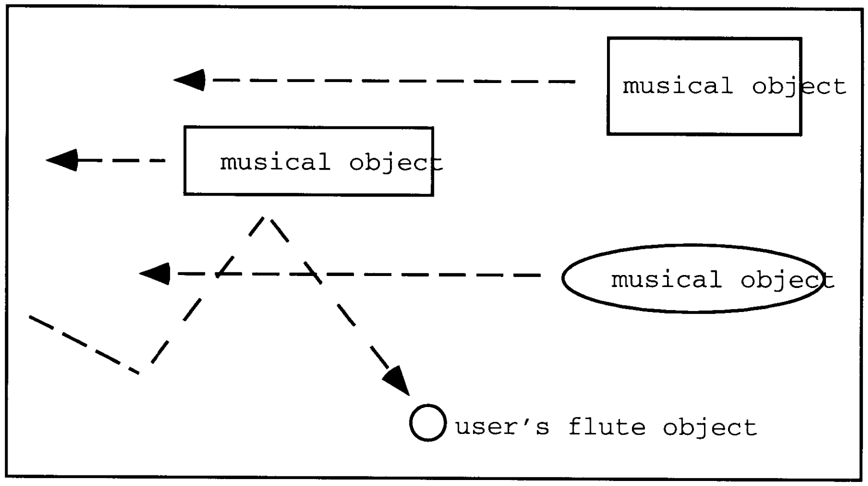
Named after the original physical interface for the project, a single transmitter-receiver pair mounted in a Barbie doll (as part of a demo for executives from Mattel Corp.), Barbie.pat is an attempt to use meaningful visual feedback for a sensor instrument.

The musical mapping is quite straightforward: hand position over the single zone is mapped directly to the pitches of a Dorian scale, played by a VL1-generated flute sound. A small circle drawn on the computer screen moves up and down with hand position, and from left to right at a constant rate.

Musical Objects

At the same time, geometric shapes representing "musical objects" scroll slowly across the screen from right to left (i.e., in the opposite direction of the little circle). By navigating the little circle (with hand position) among the geometric shapes on the screen and colliding with them, the user can hear the music that the objects represent.

FIGURE 8. Barbie.pat: visual representation (diagram).



There are three objects in the space: two simple percussion motifs (a cymbal roll and an arpeggiated bell-tree) and a complex chord object. The chord object contains a simple chord progression (vi V/ii ii V I) played by a piano. Whenever the object is struck with the circle, the subsequent chord in the progression is played.

A little Spice

When the user stops his hand in the field without withdrawing from it, a slight pitch bend is added to the flute sound (either up or down depending on the direction of approach). At the same time, the circle on the screen grows in diameter. This seemingly insignificant touch adds considerable richness and color to the instrument.

Most of the people who try this experiment find it quite entertaining and enjoyable. The ability to see what is coming upon the musical horizon, and to choose whether or not to activate a sound, is tremendously valuable for people unfamiliar with the system.

On the other hand, the experience is quite limited by its single chord progression and melodic scale. After a short period of use, one feels as if the system's musical possibilities have been exhausted. A worthwhile continuation of this experiment would be to add several additional and diverse musical objects to see how large a 'lexicon' an amateur musician could keep track of. One could even imagine creating a modal interface where certain objects would signify a change of scale or section.

Conducting: The Bach-O-Matic

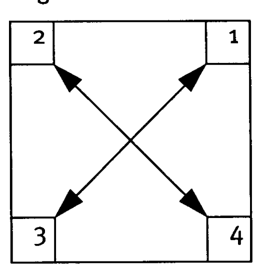
At the beginning of this research, nearly every demo of *Hand Gesture Music* was followed by the same question: "Can you use this to conduct Beethoven (Mahler, Mozart, or Frank Zappa, depending on the visitor)? My answer was always "sure, that would be easy enough, but it's not really what I'm interested in." The question gnawed at me, though, so I decided to give it a try.

For this project, I chose a Bach fugue (f# minor, Well Tempered Clavier, Book I) and recorded its four voices into four separate sequences, each with an independent control of tempo, volume, and articulation (through control of note durations).

X, Y, & Z in Transmit Mode

In this case, I felt that it would be useful to work with a real three-dimensional coordinate system (as opposed to sensor zones), so I constructed a new square sensor array and chose transmit mode which, because its response is more direct, is better suited to the task.

With the square array (i.e., four receive electrodes placed at the corners of a one and one half square foot plexi-glass vertical plate), each sensor value is treated as a vector from the center of the square to the corresponding receiver electrode:



Hand position is approximated by taking the sum of the four vectors:

$$X = \sum_{n=1}^{4} S_n \cos\left(\frac{n\pi}{2} - \frac{\pi}{4}\right)$$
$$Y = \sum_{n=1}^{4} S_n \sin\left(\frac{n\pi}{2} - \frac{\pi}{4}\right) \qquad \text{(EQ 3)}$$

Where S_n = the value of sensor n.

Harder Than it Looks

Many mappings were tried to gain comfortable musical control over the four voices, and all had limited success. At first I tried a simple mixing of the four voices. In this scenario, each corner of the space corresponds to a single voice. A hand in the middle of the space produces an equal dynamic among the voices. Moving the hand towards one of the corners makes that corner's voice louder. Various mappings of the Z dimension to articulation and global tempo were tried, but they always seemed to conflict with mixing the voice dynamics (which, even with nothing else going on, was not very satisfying).

Thus I tried higher level mappings: jagged behavior was mapped to staccato articulation and a rating of activity in general was used to stir up or slow down global tempo. I then tried linking articulation to dynamics. In each case, the latency of these mappings and their interdependencies with the parameters immediately linked to position felt completely unnatural and awkward.

Retreat

Finally, after trying a new interface with a better, more linear response, I took the advice of Professor Gershenfeld and implemented a more straightforward mapping. In this scenario, the two upper and lower voices are grouped into pairs. The Y position of the hand in the square is mapped to the voice's dynamics (i.e., a hand higher in the space makes the two upper voices louder). The X dimension is used for articulation. To the left, all notes are staccato, and to the right, all are legato, with a continuum of articulation in-between. The Z position is used to control tempo, the closer in to the plane, the faster the fugue plays.

Though this mapping produces an instrument that is far from optimal, it is at least as good (if not better) than the more complicated approaches that I attempted. This project proved so difficult because of the contrapuntal nature of the music. It is very hard, with only two arms, to control two degrees of freedom—dynamics and articulation—for four independent melodic voices. Ideally one would cluster parameters into some meaningful groupings, but, in the case of Bach Fugues wherein voices can be extremely independent, this is a very difficult musical task indeed.

Collaboration Revisited: The Sensor Frames

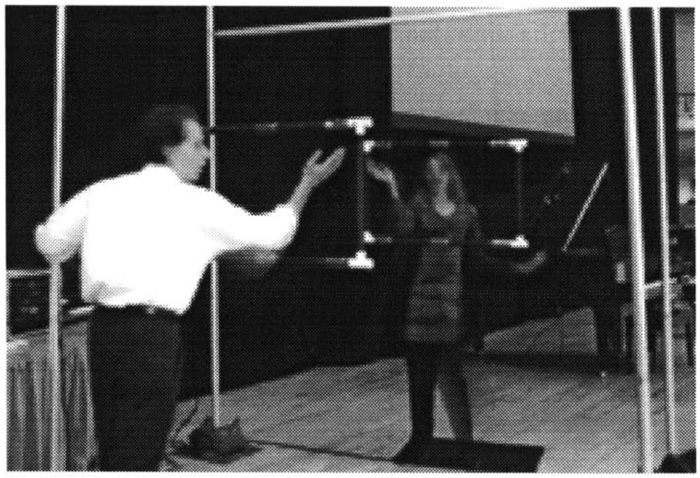
Ensemble performance

The concentration, collaboration, and extraordinary non-verbal communication inherent in playing music with others is one of the most satisfying, profound experiences that we know of, and one that is unique to music. The Gesture Cube, as noted above, was an attempt to make a system to bring this experience to amateurs. Because that experiment had many shortcomings, and because ensemble performance is so important, we revisited this theme with the Sensor Frame project.

An Interface Suited to Collaboration

One of the major flaws of the Gesture Cube project was that the interface hindered communication between the two players rather than encouraging it. When we created the single AFKAP frame, we immediately saw the new, more transparent physical interface as a potential solution to this problem. The frames project uses a pair of AFKAP-like interfaces that are played by two users facing each other. In this way, each can see the other's body, facial expressions, and every move in the active sensor space.

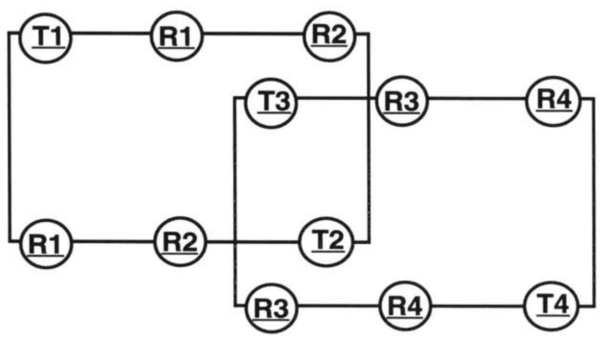
FIGURE 9. The Sensor Frames: rehearsal at Roy Thomson Hall in Toronto, 1995



Sensor Layout

Each of the frames is intended to be a two-handed instrument. To keep hand sensing disambiguated, a novel sensor layout using two Fish evaluation boards is used. Each frame employs two "half-fish" tuned to separate transmitter frequencies and forming two distinct transmitter/receiver triangles.

FIGURE 10. The Sensor Frames: configuration



The system, then, effectively consists of four independent two-dimensional sensor zones without significant cross-talk between them.

No Need For X and Y

The original idea for the frames sensor layout was that each of the four zones would output 'real' cartesian coordinates. To this, I employed a method developed by Joshua Smith to extract X and Y coordinates from a three electrode configuration [Smith 95]. Using Smith's method, I was able to calculate reliable coordinates for each zone; however, in subsequent work on the system, I found the information to be unnecessary. Raw output from the sensor zones, though not corresponding exactly to x and y in physical space, behaved similarly enough to 'real' coordinates (i.e., senors would change proportionally when a user moved his hand in a straight line across one axis) to have the same effect for the mappings I envisioned. Because information from eight sensors was already taxing the my computer system (an accelerated Macintosh IIci), I decided to opt out of the computationally expensive coordinate calculations.

New Software Tools for Togetherness

As opposed to the Gesture Cube, which depended solely on its musical content to make it an ensemble instrument, the sensor frame project has interactive mappings that *require* two players to work together in order for certain events to occur. To achieve this, it was necessary to create new software tools that correlate two user's actions and measure the extent to which they play together.

Beats_in_sync

This tool identifies when beats are played simultaneously. Whenever a beat is played in any zone, the beats_in_sync object opens a 'time-window' and looks to the other player's corresponding zone for a beat (usually the one that directly mirrors the first, but beats_in_sync can correlate any two zones). If a second beat arrives within the time window, the two are considered to be simultaneous and a sync_beat message is reported.

The E_factor

This object measures how much two player's continuous movements are similar in the sense of being each other's mirror image. To do this, the quantization of velocities for each sensor value are reduced to three states: -1 for a negative velocity; 0 for zero velocity; and +1 for a positive velocity. A collection of these velocity samples (taken at a constant rate) from a single sensor zone, represents the gesture vector associated with that zone. A similar gesture vector is measured for the second player in the sensor zone that mirrors the first, and the correlation[1] between the two vectors are computed. This correlation is, in effect a measurement of how similar the two one-dimensional gestures are. By reiterating this process for several pairs of sensor zones and averaging the results, the similarity of multidimensional gestures can be measured. The result is a number between -1 and 1 where -1 represents gestures that are exactly opposite, 0 represents gestures that are uncorrelated, and 1 represents gestures that are exactly the same (in direction and in time).

Structure

The Frames project is organized as an improvisation that consists of five interactive modes. They combine to give the piece the overall musical shape of continuous buildup and the movement from harmonic pitch to percussion to enharmonicity. Each mode also explores a different facet of ensemble playing.

Mode I: Flute Punctuations

In this first mode, each player controls a rapid staccato sequence of accelerating notes that is played on a flute timbre. When players beat together, the flute sounds are replaced by a chord of open fifths played by a harp-like sound. If players move continuously, and the E_factor is above a certain threshold, a continuous (Dorian) flute melody is played.

Mode II: Add Rhythms

This mode is similar to the first, except that the flute figure is embellished and lengthened. Also, another zone is 'activated' with percussive (backwards drum) sounds that respond to beats.

Mode III: Harps and Bells

This is a transitional mode that moves timbres towards a more rhythmic soundscape and introduces the bell timbre which is to become important in the following mode.

Mode IV: Percussion and Together Bells

This mode is the centerpiece of the Frames project. In it, each player has a solo, two-handed rhythm instrument that is complimentary to

1. The actual implementation is a variation of correlation that does not require the knowledge of these vector's dimension.

the other player's. With both instruments, the height of one hand controls the timbre of a continuous pulsing stream of percussive tones, while the other controls the speed of the pulse.

In this mode, when the E_factor is high (i.e., when user's gestures are correlated), the percussive sounds fade out and are replaced with a blurred melody of rich bell sounds.

Mode V: Bells and Long Fade

This is a concluding mode that plays a long bell resonance to end the piece.

Still Working

The Frames project is a great improvement over the Gesture Cube in terms of ensemble performance. I cannot overemphasize the extent to which the transparent (yet clearly delineated) interface and two-player mappings help in this regard. However, the piece still feels quite limited. Many frustrated attempts were made to create more diverse and interesting uses of the 'togetherness metrics.' Interesting features (like generating textures using the two player's correlated tempi) always seemed to lead to difficult instruments that required much explanation and practice to sound good. This was acceptable in the 'concert version' performed by myself and other students, but not as the public piece it was intended ultimately to be. Also, this piece once again leaves musical structure to a rudimentary sequence of static mappings.

Other Projects Using This Technology

Responsive Sound Surfaces

Michael Wu, my collaborator for the Hand Gesture Music experiment, continued his work by designing a software environment—written in the *Smalltalk* programming language—for creating interactive experiences with electric field sensors [Wu94]. For the examples presented in his thesis, Wu chose a 'shunt' mode configuration with four receiver electrodes in the corner of a square and a transmitter in the center.

Wu's approach to interaction is to have several simultaneous "layers" of detection and mapping for each sensor, creating a complex sound space to be explored by users. The majority of his examples are manipulations of timbre and textures. Though he does not attempt to create examples with temporal musical structure, he often relies on parameters that take time to develop (like "presence") in order to make textures that evolve.

Wu also points out in his evaluation that visual feedback would be beneficial to his work, noting that, when offered the opportunity, users would transfix on the monitor window that displayed raw sensor values.

The Penn & Teller Spirit Chair

By far the largest scale interactive music project involving this technology is the "Media/Medium Opera" designed and composed by Tod Machover. The opera, written for the magician duo Penn and Teller, involves many components—a musical performance, a magic trick, a narrative—at the center of which is a non-contact sensing instrument called the *Sensor Chair.* This is a particularly interesting project in that its intended users are professional and virtuosic as performers, but relatively untrained and amateur as musicians.

The Physical Instrument

The *Sensor Chair,* built by Joe Paradiso [Paradiso 95], uses 'transmit' mode with an electrode inside the seat of the chair providing the transmitter coupling, an array of four receiver electrodes arranged in a large square in front of the performer, and two sensors on the platform in front of the chair to sense foot position. The fish sensor signals were conditioned by logarithmic amplifiers to linearize sensor response, thus gaining added precision (and more effective resolution). The sensors are also equipped with lights that increase in intensity when the performer's hand approaches them (this has been very useful to amateurs who have tried the chair). Lastly, two additional sensors are used as binary switches controlled by the user's feet.

Software

The software, written in Lisp by Eran Egozy, and Peter Rice, is an adaptation for the chair of Machover's Hyperinstrument system, which organizes a piece into dynamic interactive "modes" that effectively remap user input to musical response [Machover 94b]. The parts of the opera which employ the chair's musical capabilities (several long solos as well as some shorter interludes) use many of these modes, varying from strict predetermined sequences to free improvisation, that are woven into the fabric of the larger composition.

Modes for Amateurs

Although designed as a piece to be learned and performed, several of the Spirit Chair's interactive modes have proven very successful in public presentations as instruments for novices. In one of the most popular of these, the sensor space is mapped to a 400 different drum sounds arranged by timbral quality in a 20 x 20 grid. The drum timbres play at a fine temporal quantization whenever one of the virtual grid lines is breached. Users can thus 'find' drums in the space and play them at will, or sweep their hand across the space and play the drums in a rhythm determined by the quantization.

Summary

The eight projects in which I participated as principle designer (i.e., those listed above excluding *Responsive Sound Surfaces* and the *Spirit Chair*), represent a two-year effort in making interactive music systems for amateurs using electric field sensing. I have tried to make each project an experience worthy of presentation by itself, while developing tools and ideas that will be applicable to creating such systems in general, and touching upon issues that I view as important to the development of future projects.

The course that I have charted is clearly a personal one. Choices like making a system to conduct a Bach Fugue or a Jam-session between two players, reflect my own favorite positive experiences with music and a desire to share them with others. Some of the projects, like

Barbie.pat and *MPONG* were chosen because the problems they represent (using visual feedback and making a sound-designed game, respectively) seemed so important that they had to be explored. The outcome, aside from the projects themselves, has been that my own intuition and skills in designing such systems have improved considerably. What I did not describe in the above section are the several smaller 'demo' applications, like a Calypso music player for Professor Gershenfeld, that I was able to make in the course of an afternoon by putting together tools that had been collected. The purpose, then, of the above section and the following evaluation, is to articulate the development of this expertise so that others may build upon it.

Analysis

Section Summary. In this section, we examine the examples created for this thesis in terms of their strong and weak points; present observations about designing systems with sensors; and finally, list musical generalizations: limitations and devices that have appeared in several of the projects.

Evaluations

Why This is Hard

Designing these systems always presents itself as a delicate balancing act between many interrelated factors. A system, for example, must be difficult enough—in terms of music and dexterity required—to provide a challenge, but not so hard that it becomes inaccessible to the amateur; it must be constrained enough to provide a framework for creative exploration, but not to the point of being limited and toy-like; mappings should be direct but interesting; and the system should be worthwhile both for those who play and those who observe. Indeed, one can add as many elements to this complex equation as one desires. One might want, for example, to have a system that is viable for both a ten-minute and a longer-term experience; or a system that works for crowded public installation that is also interesting for a private exploration in the home.

All of these factors must combine to meet the overarching goal of this thesis: to create a 'musically meaningful' experience for the user who plays it. This goal, as with many things that can be termed 'artistic,' is both difficult to define and much broader in scope than its compact appellation suggests. The satisfaction and joy that one feels when listening to a Haydn Symphony is quite different than what one experiences playing a four-hand reduction with a friend, or the pleasure of harmonizing a Bach Chorale. Furthermore, all of these sensations are very likely different from one person—or one day—to the next.

There is, on the other hand, *something* tangible that differentiates a 'musical' experience from, say, a theatrical one, and events in our lives *do* take on different levels of meaning. Though we cannot quantify these distinctions (at least not in the context of this document), we can use our intuition to explain what worked on the level of being musical and what didn't.

Is the music good?

This is a difficult criterion to self-evaluate, particularly in the case of examples wherein musical composition was not a primary goal (*MPONG*, the *AFKAP Frame*, and the *Bach-o-Matic*, all borrowed music and musical styles from external sources). The works which did involve original compositions (*Hand Gesture Music*, the *Gesture Cube*, and the *Sensor Frames*) all had a common problem of global structure. To give shape to a musical piece, one has to have some control over the amount of time that elements and sections take to unfold and exist before they are replaced. The amateurs who play these systems are not necessarily sophisticated enough to create larger musical form (as, for example, an experienced free-jazz improvisor would), and I was unable to find a way to impose this structure through the system. A related problem is that my particular style of composing demands a fluidity of transitions from one section to the next that I was unable to achieve without increasing the granularity of mode-changes. This inability to dovetail sections contributes to the feeling that the pieces are series of static textures rather than well-formed compositions.

The example that suffers least from these problems, not surprisingly, is *Hand Gesture Music*. This composition has the most subdivisions within its sections, and since I normally play it, its pace and timing can be carefully controlled.

I have spent a considerable amount of time and energy to make the textures rich and varied, and in this I have been more successful.

Nearly all of the sampled sounds used in the projects were recorded and trimmed specially for the works (by myself for *Hand Gesture Music* and the *Sensor Frames*, and by Adam Lindsay, Jin Kang, Maribeth Back, and myself for the *Gesture Cube*). This was clearly time well spent towards making the music feel coherent and original.

Is it enjoyable?

Almost without exception, the projects described above have been well received by test users and the public to whom they have been presented. It is clear that the demand for such instruments/experiences is high, and that, at least for now, users are willing to overlook deficiencies in systems and interfaces for the novelty of "playing" with music. Jed Smith, the president of *Cyber Smith*, a local electronic café, told me that their presentation of the Virtual Guitar described in the *Background* chapter of this thesis—and truly one of the most ill-conceived and insipid interactive musical experiences that I have ever seen—was by far their most popular interactive exhibit. Similar results have been communicated to me by other designers and presenters of interactive music systems for amateurs.

Having said that, I certainly am not contented with a "they'll like anything" evaluation of this work. There are some experiments which have been more enjoyable than others.

Observers

From the point of view of observing a system (i.e., watching a performance given by someone familiar with it), *Hand Gesture Music,* and the *Sensor Frames* have been the most popular. They are the two systems with the most sophisticated musical material and lend themselves to a certain amount of showmanship on the part of the performer(s). In the case of the *Sensor Frames*, this is enhanced by the fact that the interface and choreography of performance is visually appealing, and that the changes in musical response when two players movements are synchronized can be clearly observed.

On the other hand, these two examples are also among the most difficult to play. *Hand Gesture Music,* for instance, has been received with only luke-warm curiosity by amateurs who try it. For this piece, the complexity of mappings between user's actions and sonic response, combined with the "trick" gestures required to advance the music from section to section make it frustrating for amateurs.

Players

For "cold" attempts by amateurs at playing systems, the *AFKAP frame*, *Barbie.pat*, and the percussion mode of the *Sensor Frames* are by far the most satisfying, and those with which users seem to become the most intensely involved.

The common thread of these three experiences is that, though their mappings often produce complex results, the relationship between user's movements and musical response is direct and immediate. Thus, users can easily correlate how their actions affect the music and can quickly go beyond trying to understand the system to the point of producing results that they foresee and desire.

This leads one to favor simpler systems over the 'layers' approach (i.e., several diverse mappings detected and responded to at once), implemented in Hand Gesture Music and discussed by Michael Wu in his thesis [Wu94]. This does not, however, mean that the musical response has to be banal or simplistic. With the *Sensor Frames*, for example, the conglomerate sound of two players' rhythm instruments can become a rather complex and interesting texture. However, because mapping is direct (and, by consequence, repeatable) and immediate, users seem to grasp the system better than, for example, the *Gesture Cube*.

Is the user experience musical?

Users are not only more pleased when they can understand or follow the mapping of a system, they also play more musically: with more thought, attention, and understanding of what they are doing. As stated in the introduction of this thesis, this sort of response is at the core of our motivation for designing such experiences.

Musical systems are more than just clear ones, however; they must possess a certain quality of control that is easier to define intuitively than objectively. One criterion that I often use is whether or not two performances of a system can be distinguished and subjectively evaluated. The difference between a good and a mediocre performance is often so subtle that it is hard to put one's finger on exactly what has changed. However, if making this difference is possible given a particular system, even the most untrained amateur can sense it and feel that they are expressing themselves. Furthermore, if two experiences cannot be compared, it is impossible to improve at performing them. Without this, one cannot sustain a user's involvement for very long.

In deciding whether a system is flexible enough to be 'interpreted' in this way, it is helpful to examine what Brenda Laurel calls the variables of ***significance*** and ***range*** [Laurel 91].

Significance

Significance is the extent to which the interaction genuinely effects the system. Are the user's choices, as Laurel puts it, "cosmetic," or do they really affect something. For example, to start the background groove in the *AFKAP Frame*, one only has to approach the sensor field. Despite its simplicity, users realize that this control is 'real', understand it, and often produce musical results (for example, "playing DJ" by starting and stopping the groove on a beat). I would add two qualifications to Laurel's description:

1. Significance is only valuable to an amateur user if it can be perceived. For example, the layered approach of *Hand Gesture Music* is so complex that, even though gestures do indeed shape the music, it is impossible for some users to understand how this occurs and therefore is, in effect, no more 'significant' to them than listening to a sequence.
2. Significant and perceptible control does not, on the other hand, have to be blunt control. This is especially the case for music systems which <u>ought</u> to attune users to the level of nuances. This is difficult because subtle control can often cross the line and become confusing.

Range

The on-groove/off-groove interaction with the *AFKAP Frame* is, of course, only a binary choice, and cannot hold up in the long-term because one's options are so quickly exhausted. ***Range*** is the measure of a user's significant options, and is a necessary element in making systems that are 'interpretable.' *Barbie.pat* offers a slightly higher range than the *AFKAP Frame* in that users can choose what notes of the Dorian scale to play and when to collide with musical objects. Still, however, it sounds more or less the same no matter what choices one makes. The sensor Frames, on the other hand—especially in its full version—offers many more choices: whether or not two players mirror each other, take solos, make synchronous beating gestures, etc. Consequently, performances can vary widely from flat and boring, to exciting and 'in the pocket.'

Explicit Constraints

It would be nice to say, "well, just turn up the 'range' and 'significance' knobs and you'll have the perfect, meaningful system." To do this we need only to hand each user a violin and call it a day. This would, clearly, be too difficult for an amateur. A balance must be struck between making systems that allow interpretation and keeping them constrained enough so that amateurs can play them. Even a professional violinist is likely to have difficulty producing anything worthwhile if asked, without a score in mind or on paper, to "play something." He is much better off with constraints like, "improvise on this chord progression," or "play that Beethoven Sonata." Amateurs need to have constraints even more so than professionals. Most importantly, designers must somehow make these constraints explicit without the benefit of experienced performers and musical notation to help them.

Learning Curve

In first designing interactive experiences, and even in writing the proposal for this thesis [Waxman 94], my intuition led me to strive for systems that were completely self-explanatory. A user, I thought, should be able to start with a system without any help from a person or instruction booklet, or even any prior knowledge about what might happen to them during the experience. Furthermore, an instrument with an "ideal" learning curve would allow users to discover its idiosyncracies and achieve continually more interesting musical results with time and practice.

Though this is clearly still a seductive ideal, pragmatic experience has shown that a little instruction can go a long way towards making the experience feel more satisfying and profound. In other words, the payoff of planning for a ten minute learning curve, even to get off the ground, as opposed to a thirty second one, is well worth the user's time and presenter's effort if it is possible (unfortunately, in most public installations it is not).

For example, in designing the *Gesture Cube*, we intended the entire experience to be self-explanatory except for one gesture: to change sections both users had to put both hands all the way into the field and then step completely out of the field. It was a gesture that had to be taught. The reason, I admit, for this compromise, was that we couldn't think of any other way that would not cause accidental section changes or be impossibly confusing for the users. One could argue, however, that some of the most successful and magical moments in playing the cube were achieved with this intentional, choreographed gesture. It was a pleasurable for people to learn a 'technique' and rise to the little challenge of 'performing' it.

Good Difficulty

In general, then, a little bit of difficulty might not be so bad for amateur systems. A user should be encouraged to think, pay attention, and perhaps even learn a little about how a system works. However, his efforts must be rewarded with output that shows that he has done something right. Once again, this reward can only be valid to the user if he can deem one performance to be better than another. Furthermore, if one expects a user to improve at playing a system by learning gestures, timings, or sequences of actions, these must produce repeatable results. For example, for *Hand Gesture Music* and the *Gesture Cube*, I often mapped a simple beat to the launch of a complex, sometimes randomly generated sequence of note events. This was very effective in creating an interesting texture out of a simple input, but proved frustrating to play because one quickly realized that events were out of control and that no amount of practice could reign them in. The drum mode of the *Sensor Frames* also generates complex results from simple mappings: a continuous pulse is modulated in tempo by one hand, and in timbre by the other. Despite this complexity, a two identical gestures produce identical output, and thus the one feels as if the system is under control and that one can improve at playing it.

Observations

Gaming

One novel way to approach these systems is to add constraints and 'good difficulty' as game-like challenges. *MPONG* and *Barbie.pat* are simple examples of this. Both offer a visual component that represents a challenge separate (but not divorced from) the musical output of the system. In *MPONG*, the object is: "Keep ball moving wildly for best results." In *Barbie.pat*, the it is simply to collide with musical 'objects' when one wants to hear them. While hardly a game in the traditional sense, this small reward of having intention verified with results made the experience far more successful than it would have been otherwise. In any case, the challenge of 'achieve this', as opposed to the vague dictum 'make music,' is a very useful distinction to make.

Non-contact Sensing and Music

Unencumbered Movement and Sound

Making music by gesturing in an open space allows for a freedom of movement that is often quite exhilarating. One of the things that I learned in playing with these experiences is that they create an interesting feedback loop between controlling a system and 'dancing' in response to the sound it produces. Indeed, when I first began giving demonstrations of *Hand Gesture Music*, some of my colleagues remarked on how much I seemed to be "acting" as I performed. When playing a section in which beat triggers control large gong sounds, I tended to use large swings of the arm, whereas when playing the delicate bell tree sounds that come at the end of the piece, I created beats with a twiddle of the fingers. While the two gestures are identical as far as *Hand Gesture Music's* rudimentary recognition software is concerned, I was not acting. In fact, it is as if I could 'feel' the instruments that I was playing.

Music Without Tactile Feedback

In spite of this imagined 'physicality,' the lack of any real tactile feedback is often a serious drawback. Without it, the user is denied an important aid in refining his gestures and repeating the movements that he has learned. With DBX II, for example, users are able to use minute alterations of Joystick position to control equally minute changes in the music produced by the system. The joysticks provide not only something to grasp, but a point of reference that would be absent if the program used Fish sensors for its interface. To use a common musical term, non-contact sensing does not facilitate the development of "finger memory."

In addition, playing gesture sensing instruments can sometimes leave users feeling awkward and 'exposed,' with no object upon which to focus. If a user's attention drifts from the musical sounds (or the other player in the case of two person instruments), he can easily come to the self-conscious and intimidating realization that he is waving his arms in the air. Lastly, one simply misses the tactile pleasure that traditional instruments offer. It feels good to cradle one's 'cello or lean into the keys of a piano.

Element of Magic

The idea that one can move ones hands and produce sounds has thus far had great novelty-appeal to users unfamiliar with the technology. One wonders, however, if this magical element will have lasting value as non-contact sensing becomes widely available in more mundane products like three dimensional computer mice and kitchen appliances.

Visual Feedback

Though tactile feedback is not possible with the current technology, visual feedback is, and the two experiences which employ it—MPONG and Barbie.pat—have a particular quality that none of the other experiences have. Like the Joysticks used in DBX and mentioned above, visual feedback can give users a point of reference which allows them to refine their gestures and achieve a greater sense of control.

Indeed, visual feedback can provide a point of reference in time as well as in space. One of the problems with making these systems is that amateurs cannot read musical notation. For this reason, music is often limited to the moment, to static sound environments that can be explored. In order to give shape to music over time, user's must be given a sense of 'what's coming' and make decisions about musical development. The visuals employed in *Barbie.pat*, though simple, are a step in the direction of providing an alternative 'notation' can be employed to this end. Users who try Barbie.pat quickly learn the musical reactions of the three objects with which they collide, and make decisions about whether or not to hit them with the sonic outcome already in mind. A continuation of this research would elaborate on this rudimentary notation, both in the number and complexity of objects (some objects had 'states' such as color and vertical position which were identifiable but not exploited to reflect their content) to see how far this prediction of outcomes can be sustained.

Musical Generalizations

Although every project detailed above has had its own distinct musical content, it is possible to make a few generalizations about composing for non-contact sensing and amateurs.

Precision of Control

Continuous types of control, such as changing a timbre color, work more effectively than control requiring precision, like choosing pitches to play a melody. Rhythmic precision (using beats) is easy to detect but difficult for amateurs to accomplish without the help of quantization. Generalizations about behavior, such as whether a person is playing jaggedly or quickly, are also rather easy to detect, but these features take some time to unfold. Thus, the musical response to such behaviors cannot be immediate, and this makes mappings difficult to comprehend. In the *Bach-o-matic*, 'cumulative' features like these are used to control tempo and articulation with some success. However, when users discover these mappings, they tend to play them to the detriment of other mappings (such as voice mixing) which are controlled at the same time and in the same sensor space.

Repetition and Loops

One solution to the ever-present problem of creating musical structure has been to create textures that evolve but repeat so that users come to know them and 'remember what will happen next.' Many of the above projects involve loops and repetition (MPONG is the best example, using a traditional repeating blues progression). Though this should not be an end-all solution, it is a tried and true one that has proven itself in most forms of Popular and Folk music.

Color and Textures

Many of the above examples depend on manipulating timbre and textural color rather than rhythmic layers or melody. This has been achieved with a heavy dependance on pitched and sustained percussion (such as bells and gongs), pedal tones, and 'spectral' harmonies which change internally more than they progress harmonically. These devices tend to hold up well to the imprecision of the sensors and the amateur users who play them.

Easy Music

When users are asked to control pitch—like in *Barbie.pat*—it is sometimes advantageous to give them melodic material that is tonal and familiar. Restricting notes to Dorian or Pentatonic scales, which are forgiving to the exigencies of voice leading, has been a good way to make melodic material that is controllable. Another way to chose notes using imprecise control is to do so randomly from a collection of pitches (often with weights given to more 'important' ones) that sound together as a harmony. This technique has been also been encouraged by the Max environment which is exceptionally well suited to this type of process. On the other hand, depending entirely on scale quantization or pitch collections creates a music with no 'bad notes.' When employing these devices, one must find other means to allow the user to generate music that can be subjectively evaluated, such as control of dynamics or phrasing.

Future Work

Section Summary. This section suggests improvements that could be made to the work both in the short and long term, and outlines several directions that future work in this area might take.

Gesture Sensing

There are several ways in which improvements in gesture sensing technology, both hardware and software, could contribute to the creation of better interactive experiences for amateurs.

The Smart Fish

One forthcoming improvement is a new generation of Fish sensors, called 'SmartFish'. These will have more sensor channels (eight), more resolution (16-bit), and an on-board digital signal processing chip (Analog Devices) that will automate many of the tasks that have thus far been either manual (like calibration) or handled by software running on another computer (like calculation of position) [Zimmerman 95].

These improvements will allow us to have sensor systems that are more accurate and easier to set-up and calibrate. The example of Tod Machover's *Spirit Chair,* which uses custom software and hardware to achieve similar goals, shows that such improvements are indeed valuable. Mappings can be made to be more repeatable, both from one person and one place to the next. One cannot imagine creating systems for commercial or home use without the benefit of auto-calibration.

Gesture Sensing Software

The new sensing hardware will also facilitate the development of improved gesture sensing software. With the current software, we are limited to detecting rudimentary positions and a few general behaviors like jaggedness or velocity. Ideally, one could imagine detecting more complex composite gestures (more akin to sign language) in order to create a 'lexicon' that would allow users to communicate with systems on a much more specific level than they can presently. Imagine, for example, being able to point to individual instruments in a 'virtual' orchestra, selecting them with one hand and controlling their articulation and volume with the other.

Imaging software already in progress by Physics and Media student Joshua Smith [Smith 95], will be able to detect the shape of an object in a sensor field as well as its position. The ability to differentiate a clenched fist from a pointed finger will be of great aid in the development of the aforementioned 'gesture lexicon'.

Force Feedback

As we stated in the *Analysis* section of this work, one of the drawbacks of non-contact sensing is the lack of force feedback. Force feedback devices such as the Phantom Haptics, Inc.'s *Phantom* and the *Modular Feedback Keyboard* [Cadoz 90] are currently being developed and improved. A logical step in this research would be to use one of these devices as an interface to music, and explore what a sense of touch can genuinely add to an amateur user's experience.

Visual Feedback

One of the most interesting projects created for this thesis was *Barbie.pat*, I believe due to its use of visual feedback that gives more information to the user about where his actions fit in a larger timescale. These visuals, however, were rudimentary and did not fully explore what one might be able to achieve in terms of user involvement and understanding of a complex system. Clearly this is a subject worthy of a thesis of its own in the near term. Tod Machover's work on musical games for the *Brain Opera* [Machover 94a], and recent projects by John Underkoffler which allow users to 'fly' through a three dimensional representation of a musical score, show tremendous promise in this area.

Getting to Know the User

In creating this thesis, I have made quite a few assumptions about what it takes to create 'musical' experiences for amateurs. Though several of the projects were presented publicly, I regret not having collected more systematic data about people's reactions, difficulties, and ultimately, what they felt they were getting out of the experiences. It would be extremely beneficial to the development of work like this to 'user-test' systems (either with video cameras, questionnaires, or well-documented observation) in order to discover whether any musical elements, mappings, or organizations prove successful on more than a case by case basis. One example of a mistaken assumption was the use of beat gesture detection for the *Gesture Cube*. Snapping my wrist in the air to signify an ictus seemed completely natural and intuitive in *Hand Gesture Music*. About half the people who played the *Gesture Cube*, however, would make beats with a much larger and more rigid gesture of the arm. Eventually, I reduced the system's velocity threshold so that both beats would work. I might never have discovered this phenomenon, however, showing the system to a handful of users in the studio.

Integrated Public Spaces

The Brain Opera

Making experiences simple enough to be played by inexperienced users visiting a public space for a short period of time, yet complex enough to be interesting and meaningful is a recurring paradox. One way to confront this is to construct a larger-scale artistic experience out of a collection of simpler, related mini-experiences. Tod Machover's current project, the *Brain Opera* [Machover 94a], takes this approach. The project, slated for its first performance at New York's Lincoln Center in the Summer of 1996, will combine amateur music games, interactive group experiences, and performance in a progression of increasing intensity and involvement. The opera derives its whole-from-parts structure from its theme: the inner-workings of the human brain as described by Marvin Minsky in his book, *The Society of Mind* [Minsky 85].

virtual spaces

An aspect of the *Brain Opera* that I find particularly fascinating is that some of its 'performance spaces' will be virtual ones. The *Virtual Brain Opera* will bring users together on the Internet and allow them to take part in the experience from wherever they are connected. One of the advantages of using the Internet is that people can take part in the experience for as long and as often as they like, and will thus have a chance to become intimately familiar with the music and thinking behind the work. From the perspective of those who see the 'physical' *Brain Opera*, the contribution of those on the Internet will give the experience a sense of expansiveness and continuity that it would not otherwise have.

Conclusions

The eight projects described in this thesis represent two years of experimentation with interactive music systems for amateurs using electric field sensing. Though the projects are in some ways disparate, they are no more so than the broad span of musical experiences they try to emulate: from group experiences to personal ones; improvisation to thoughtful execution of a great composer's work. The common thread is that they all try and bring an experience that is 'musical'—defined perhaps by my own various relationships with music as a composer, performer and listener—to a public that would not otherwise know what those experiences are like.

In thinking toward the future, I return to Max Matthews vision of a new listener who buys his/her music as a computer program rather than a compact disk or cassette. I stop short of this vision in that I don't believe any of these systems will replace existing means of relating to music. On the other hand, systems like these will likely go far beyond simply providing easier and more accessible ways for amateurs to listen to music. After building these projects, I realize that composers cannot simply translate their music to new electronic 'instruments,' but must stretch their own thinking to express themselves in a completely new medium. Over the last two years I have improved as a composer of interactions as much as a composer of music, and though one can never ignore the latter, the two cannot be thought of independently.

I can only look forward to what might come of this search for expression in the interactive medium: experiences yet to be imagined that are as profound and varied as those musical ones which we enjoy today.

Bibliography

[Barrière92] Jean-Baptiste Barrière, "The Creative Ambiguity", from *Revue Virtuelle, Notebook 2-3*; trans. Paul Smith. Éditions du Centre Pompidou, Paris 1992.

[Behrman 91] David Behrman, "Designing interactive computer-based music installations," *Contemporary Music Review*, vol. 6, Part 1; Harwood Academic Publishers, 1991.

[Blacking 87] John Blacking, *A Commonsense View of All Music*; Cambridge University Press, Cambridge U.K., 1987.

[Boulanger 90] Richard Boulanger, "Conducting the MIDI Orchestra, Part 1: Interviews with Max Mathews, Barry Vercoe, and Roger Dannenberg," *Computer Music Journal*, Vol. 14, No. 2, Summer 1990, Massachusetts Institute of Technology Press.

[Boulez 81] Pierre Boulez, *Repons*, Musical Score, 1981.

[Burgh 1814] A. Burgh, *Anecdotes of Music, Historical and Biographical, in a Series of Letters from a Gentleman to his Daughter*; from: Piero Weiss and Richard Taruskin, eds.*Music in the Western World: a History in Documents*; Schirmer Books, New York, 1984.

[Cadoz 90] Claude Cadoz, Leszek Lisowski, and Jean-Loup Florens, "A Modular Feedback Keyboard Design", *Computer Music Journal*, Vol. 14, No. 2, Summer 1990, Massachusetts Institute of Technology Press.

[Chadabe 91] Joel Chadabe, "About M," *Contemporary Music Review*, vol. 6, Part 1; Harwood Acedemic Publishers, 1991.

[Clynes 77] Manfred Clynes, Sentics, *The Touch of The Emotions*; Doubleday, New York 1977.

[Darter 84] Tom Darter, ed., *The Art of Electronic Music*, GPI Publications, 1983.

[Eskew 86] Harry Eskew and James C. Downey, "Shape-note Hymnody," from*The New Grove Dictionary of American Music*, v4. Wiley Hitchcock and Stanley Sadie, eds. Macmillan, New York 1986.

[Gabriel 93] Peter Gabriel, *Xplora*, Peter Gabriel Ltd. license to Real World Multi Media, 1993.

[Galeyev 91] Bulat M. Galeyev, "L.S. Termen: Faustus of the Twentieth Century",*Leonardo*, Vol 24. No.5. Pergamon Press, 1991.

[Gehlhar 91] Rolf Gehlhar, "SOUND = SPACE: an interactive musical environment," *Contemporary Music Review*, Vol. 6, Part 1; Harwood Academic Publishers, 1991.

[Machover 92] Tod Machover,*Hyperinstruments: A Progress Report*, 1992.

[Machover 94a] Tod Machover,*Brain Opera and Virtual Brain Opera*, project proposal, MIT Media Laboratory, 1994.

[Machover 94b] Tod Machover,*Magic Mini Opera*, project description, MIT Media Laboratory, 1994.

[Machover 94c] Tod Machover, program notes from Professor Machover's class project:*Vox-Cubed*. MIT Media Laboratory, 1994.

[Mathews 80] Max Mathews, *The Sequential Drum*, Rapports IRCAM, Paris, 1980.

[Matsumoto 93] Fumiaki Matsumoto, *Using Simple Controls to Manipulate Complex Objects: Application to the Drum-Boy Interactive Percussion System*, M.S. for MIT Media Laboratory, 1993.

[Minsky 85] Marvin Minsky, *The Society of Mind*; Simon and Schuster, New York 1985,

[Paradiso 95] Joe Paradiso and Neil Gershenfeld, "Musical Applications of Electric Field Sensing," to be published, 1995.

[Puckette 90] Miller Puckette and David Zicarelli, *Max Development Package*, Opcode systems, 1990.

[Rigopolus 94] Alexander Rigopolus, *Growing Music from Seeds: Parametric Generation and Control of Seed-Based Music for Interactive Composition and Performance*; M.S. for MIT Media Laboratory, 1994.

[Rowe 93] Robert Rowe, *Interactive Music Systems*; MIT press, Cambridge, MA, 1993.

[Sessions 50] Roger Sessions, *The Musical Experience of Composer, Performer, Listener*; Princeton University Press, Princeton, NJ, 1950.

[Smith 95] Joshua Smith, *Toward Electric Field Imaging*, SM thesis, MIT Media Laboratory, Cambridge, 1995.

[Waxman 90] David Waxman, *Eléments d'Analyse: Pluton*; Cahiers d'Analyse Technique IRCAM, Paris, 1990.

[Waxman 93] David Waxman, "An Interactive Exporation of Computer Music Research"; *Computer Graphics Visual Proceedings*, ACM SIGGRAPH publications, 1993

[Waxman 94a] David Waxman, "Participatory Music Systems for Non-Expert Users"; internal document, MIT Media Laboratory, 1994

[Waxman 94b] David Waxman, *Theme and Interactions: a Participatory Suite*, M.S. thesis proposal, MIT Media Laboratory, 1994.

[Wu 94] Michael Wu, *Responsive Sound Surfaces*, M.S. for MIT Media Laboratory, 1994.

[Zimmerman 95] Thomas G. Zimmerman, Joshua Smith, Joseph Paradso, David Allport, and Neil Gershenfeld, "Applying Electric Field Sensing to Human-Computer Interfaces"; proceedings of CHI'95.

[Zimmerman 95b]Thomas G. Zimmerman, *Personal Area Networks (PAN): Near-Field Intra-Body Communication*, SM thesis, MIT Media Laboratory, Cambridge, 1995.

Appendices

A: Video Examples

A video cassette of the projects described in this thesis is available from the MIT Media Laboratory Hyperinstruments group. For more information, please contact:

Susan Bottari

Massachusetts Institute of Technology

Room E15-495A

20, Ames Street

Cambridge, MA 02139-4307

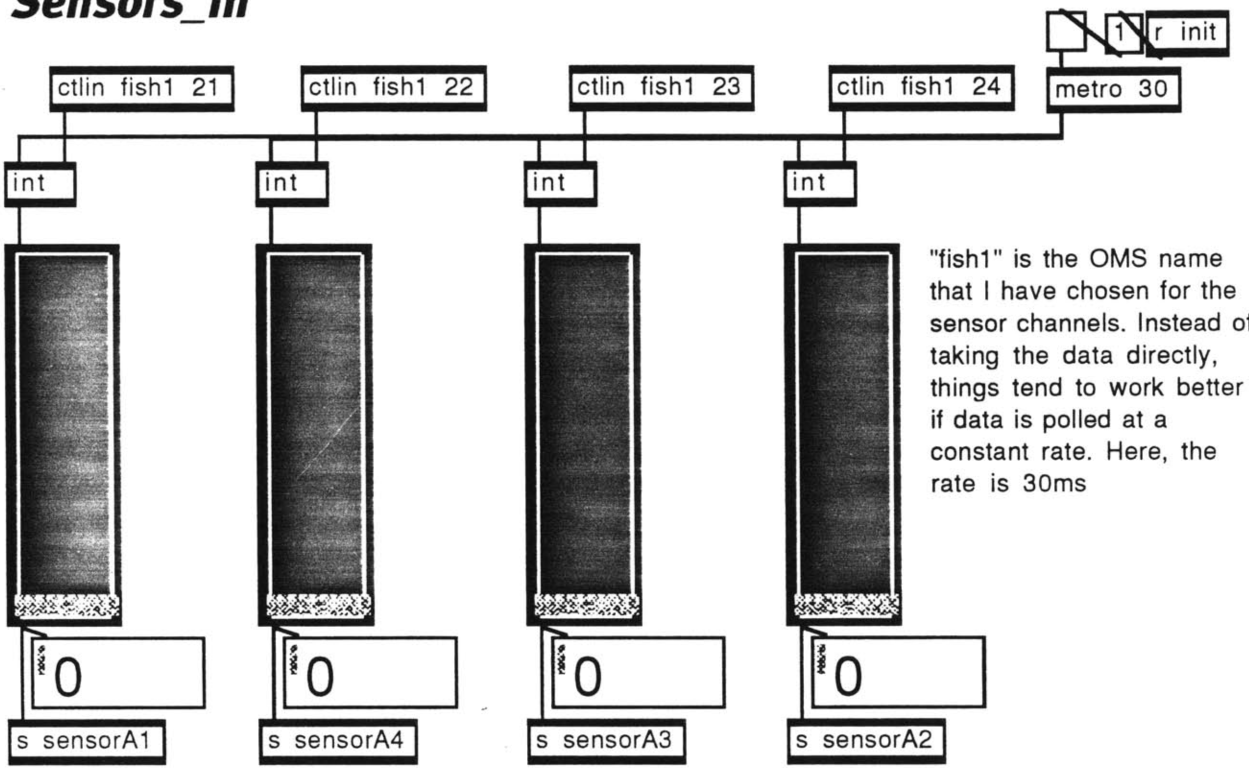
Tel: (617) 253-0392

Fax: (617) 258-6264

B: Example Max Patches

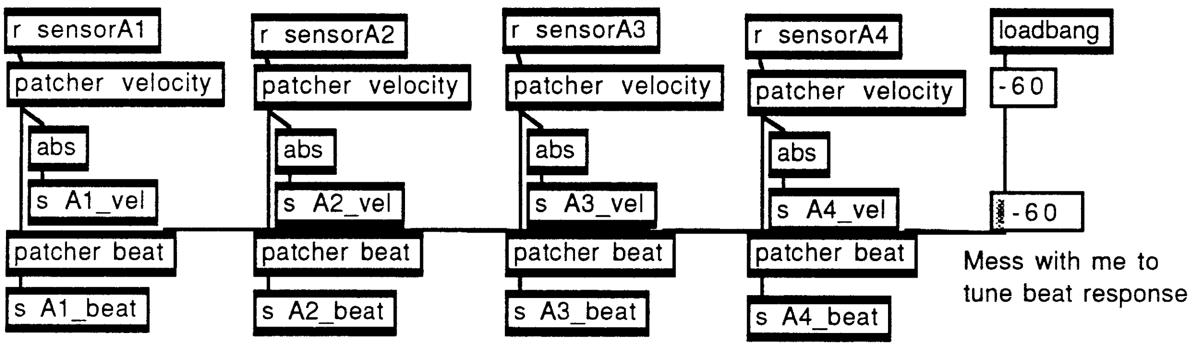
Following are printouts of the 'software tools' described in the 'Projects' section of this document.

Sensors_in

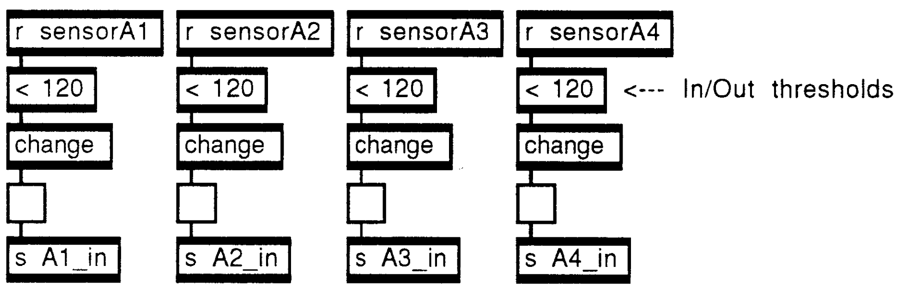

r init
1
metro 30
ctlin fish1 21
ctlin fish1 22
ctlin fish1 23
ctlin fish1 24
int
int
int
int
0
0
0
0
s sensorA1
s sensorA4
s sensorA3
s sensorA2
"fish1" is the OMS name that I have chosen for the sensor channels. Instead of taking the data directly, things tend to work better if data is polled at a constant rate. Here, the rate is 30ms

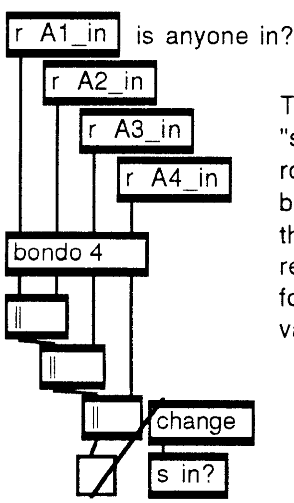
Gesture_rec

Beats



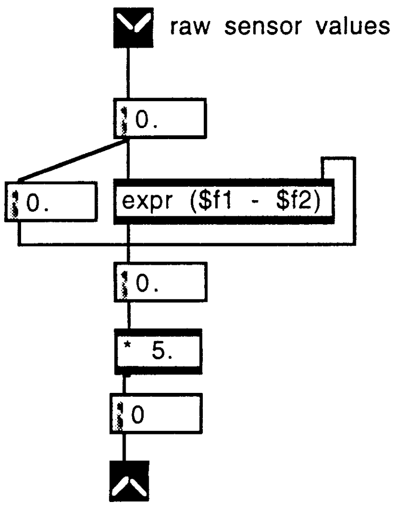
In/out



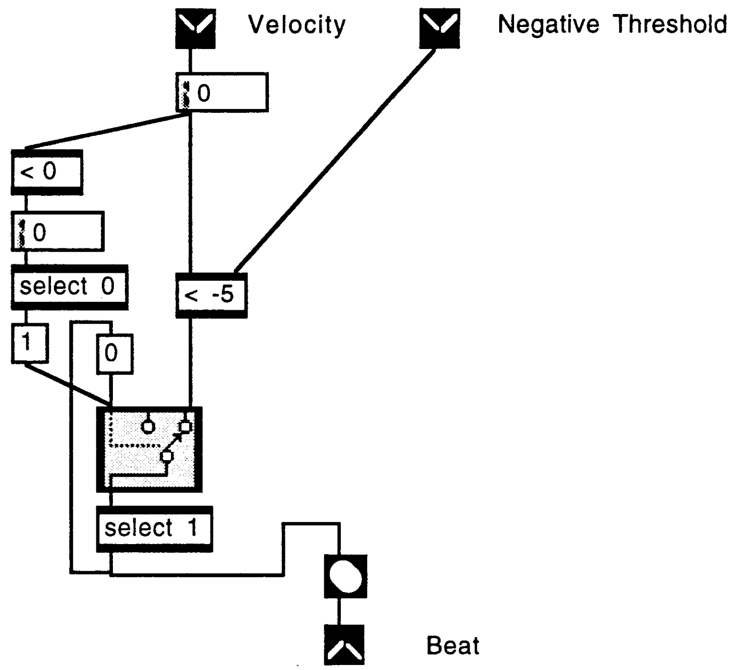


This patch detects some simple activity in the sensors. "sensorA1...A4" give raw values for each receiver. This also roughly corresponds to Z direction. A1...A4_beat reports a bang when a beat (or rapid change of velocity) occurs in one of the sensor regions. A1...A4_vel is velocity. A1...A4 simply reports whether or not there is a hand detected in one of the four sensor fields as determined by a threshold. The threshold value can be set as desired.

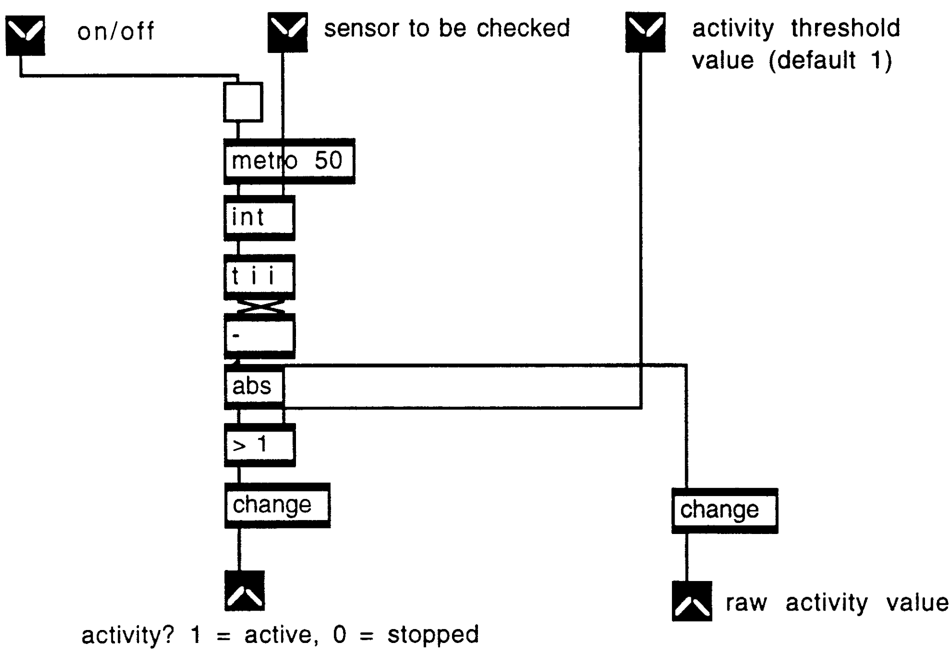
velocity



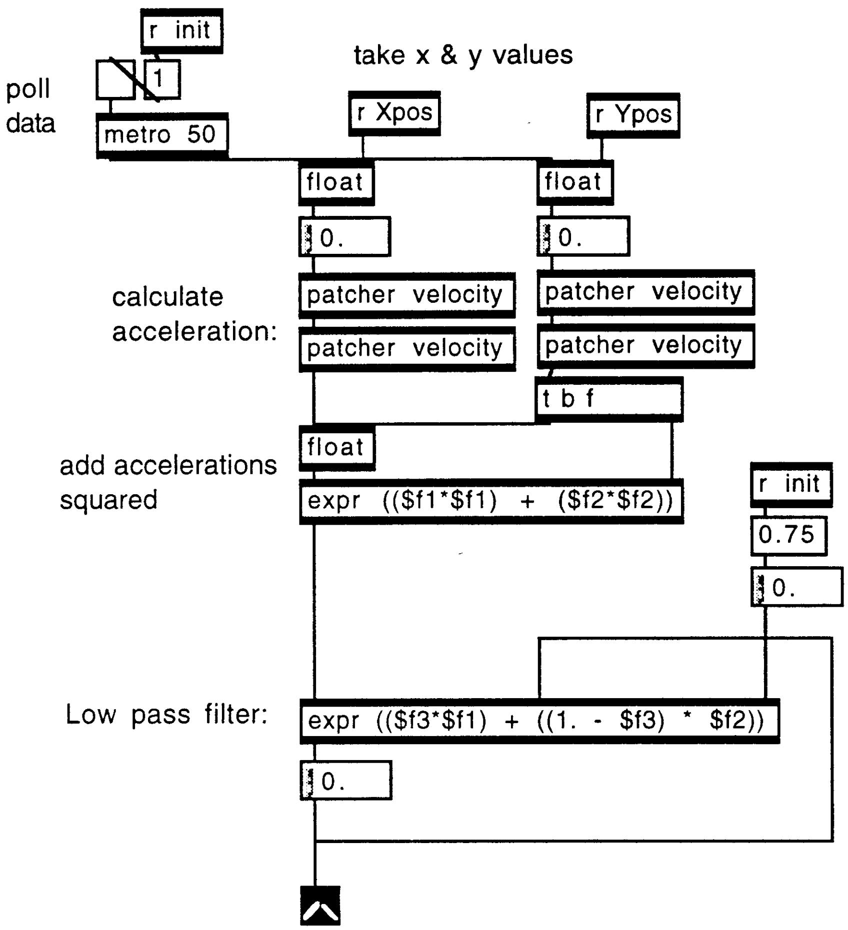
beat

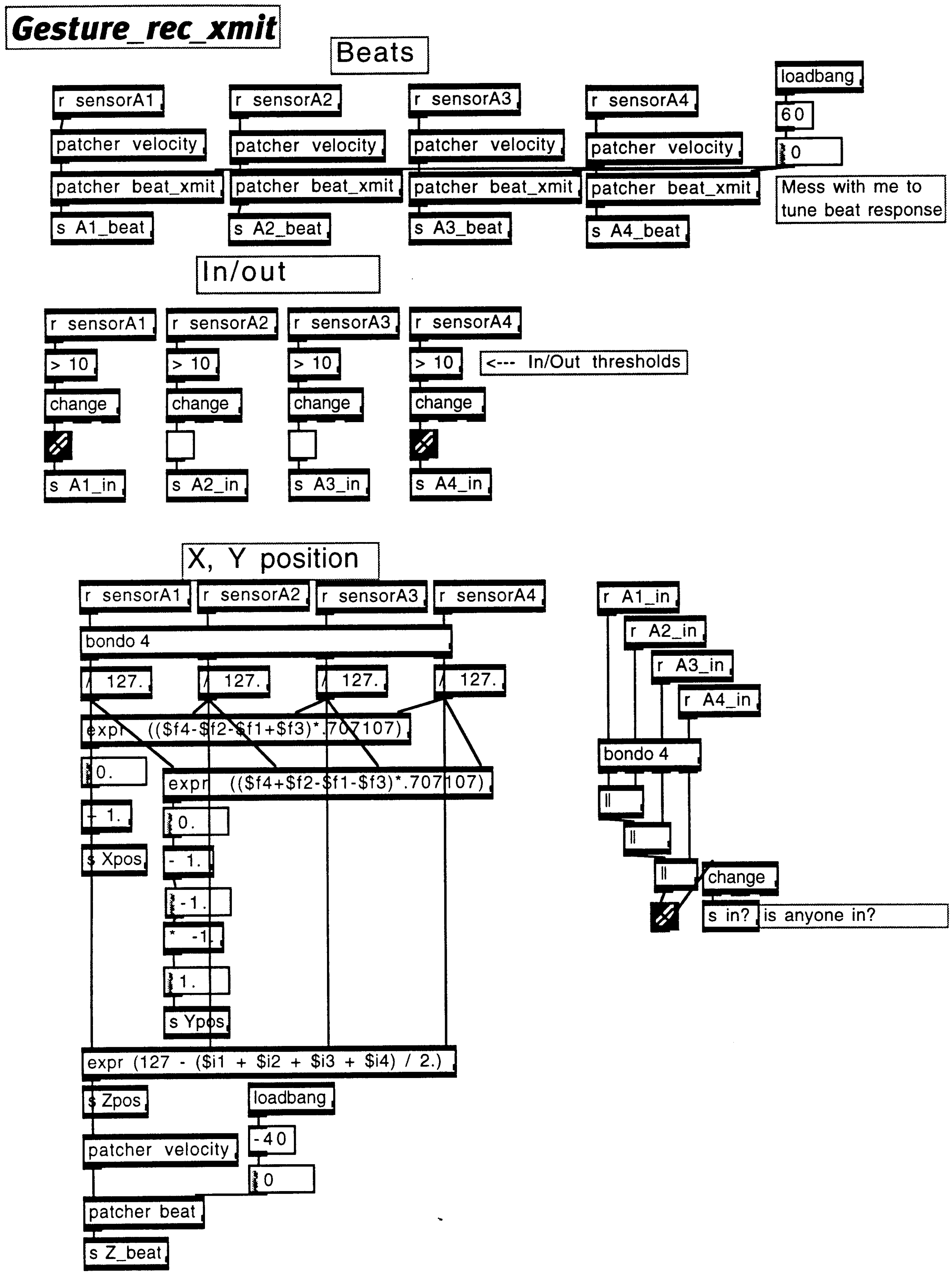


active?

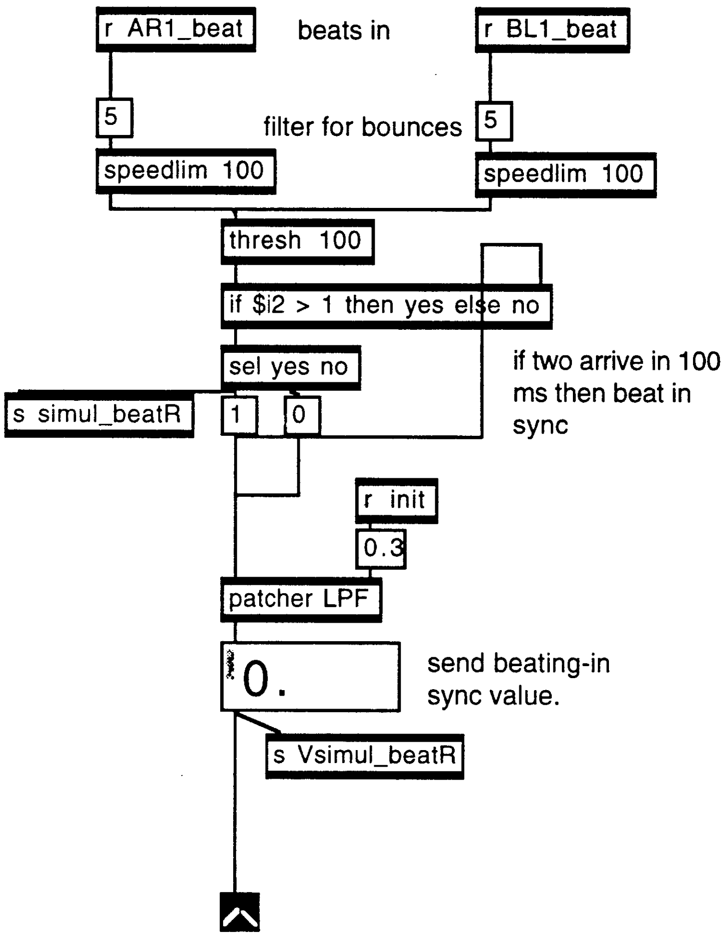


Jagged, Acceleration, and Low-Pass

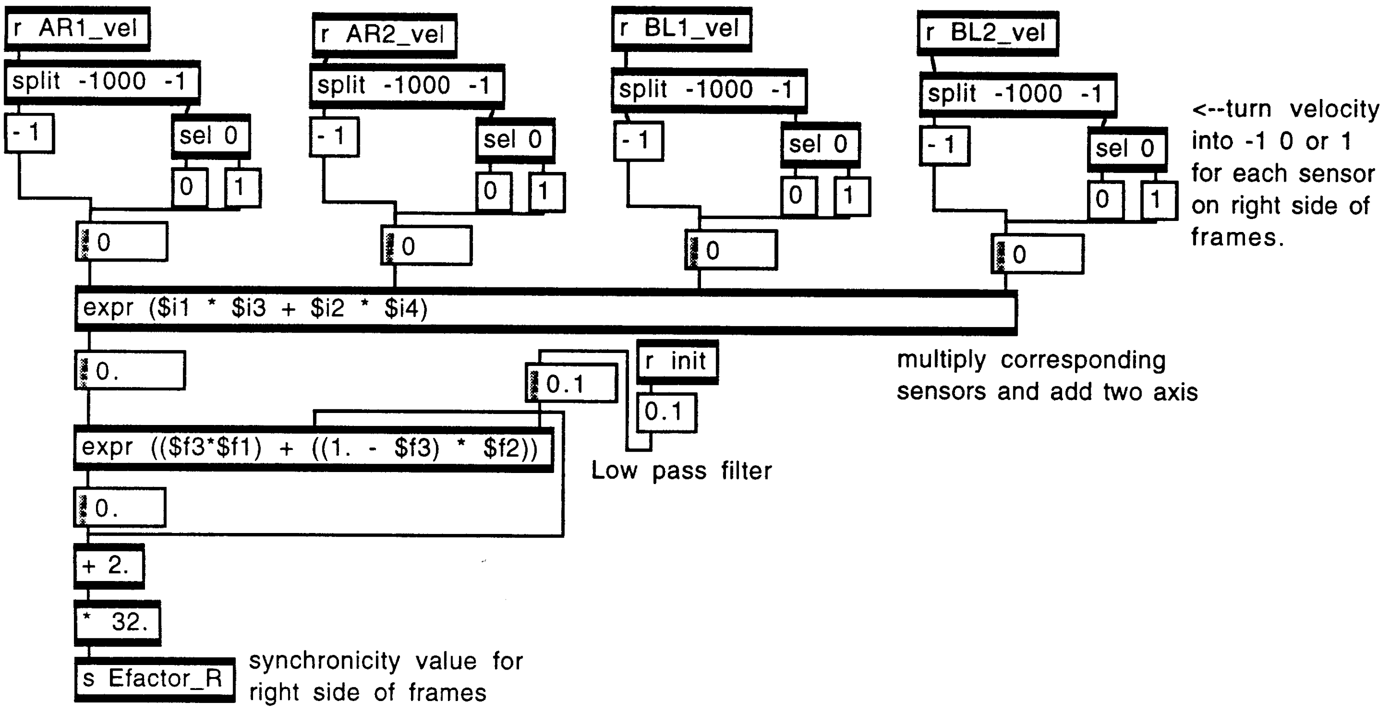



Gesture_rec_xmit
Beats
loadbang
60
0
Mess with me to tune beat response
r sensorA1
r sensorA2
r sensorA3
r sensorA4
patcher velocity
patcher beat_xmit
s A1_beat
s A2_beat
s A3_beat
s A4_beat
In/out
> 10
<--- In/Out thresholds
change
s A1_in
s A2_in
s A3_in
s A4_in
X, Y position
bondo 4
/ 127.
expr ((\$f4-\$f2-\$f1+\$f3)*.707107)
expr ((\$f4+\$f2-\$f1-\$f3)*.707107)
0.
+ 1.
s Xpos
- 1.
-1.
* -1
1.
s Ypos
expr (127 - (\$i1 + \$i2 + \$i3 + \$i4) / 2.)
s Zpos
loadbang
-40
0
patcher velocity
patcher beat
s Z_beat
r A1_in
r A2_in
r A3_in
r A4_in
bondo 4
||
change
s in?
is anyone in?

Beats_in_sync

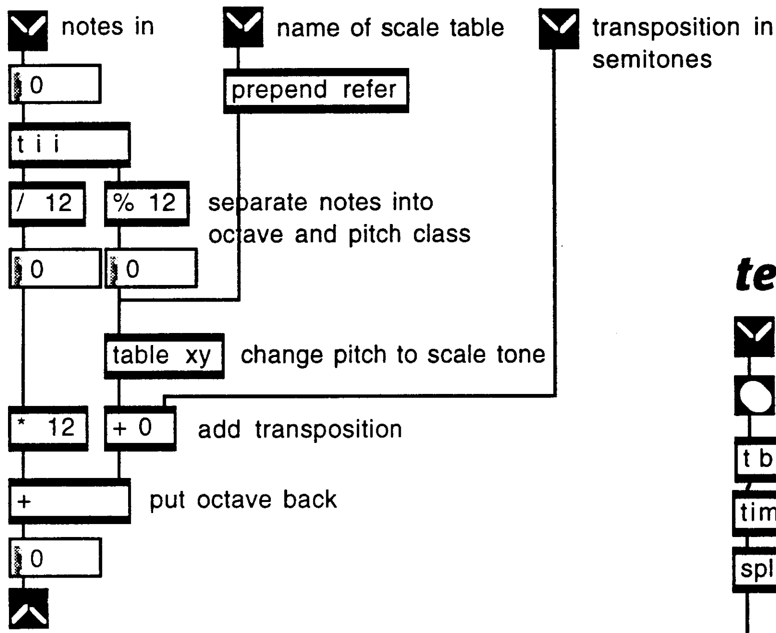


the E_factor

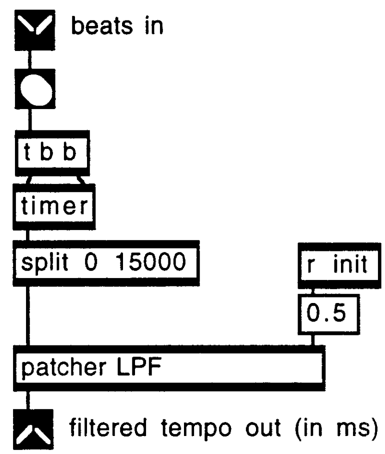


r AR1_vel
r AR2_vel
r BL1_vel
r BL2_vel
split -1000 -1
- 1
sel 0
0
1
<--turn velocity into -1 0 or 1 for each sensor on right side of frames.
expr ($i1 * $i3 + $i2 * $i4)
0.
0.1
r init
multiply corresponding sensors and add two axis
expr (($f3*$f1) + ((1. - $f3) * $f2))
Low pass filter
+ 2.
* 32.
s Efactor_R
synchronicity value for right side of frames

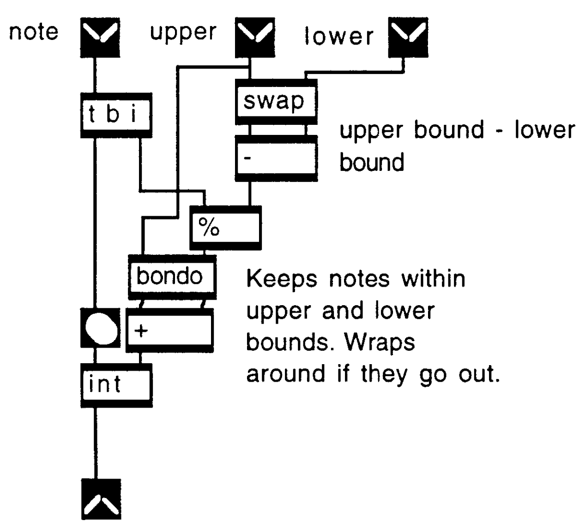
trans_scale



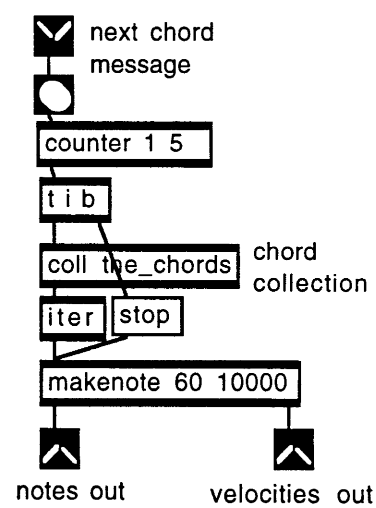
tempo_maker



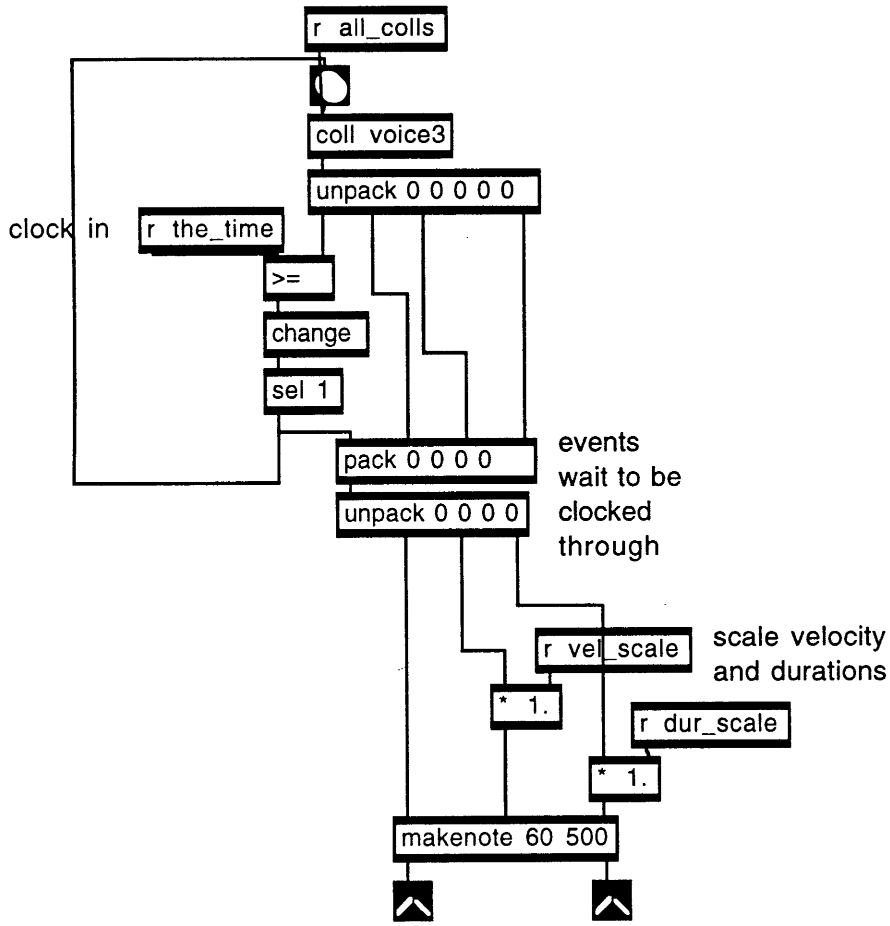
ambi



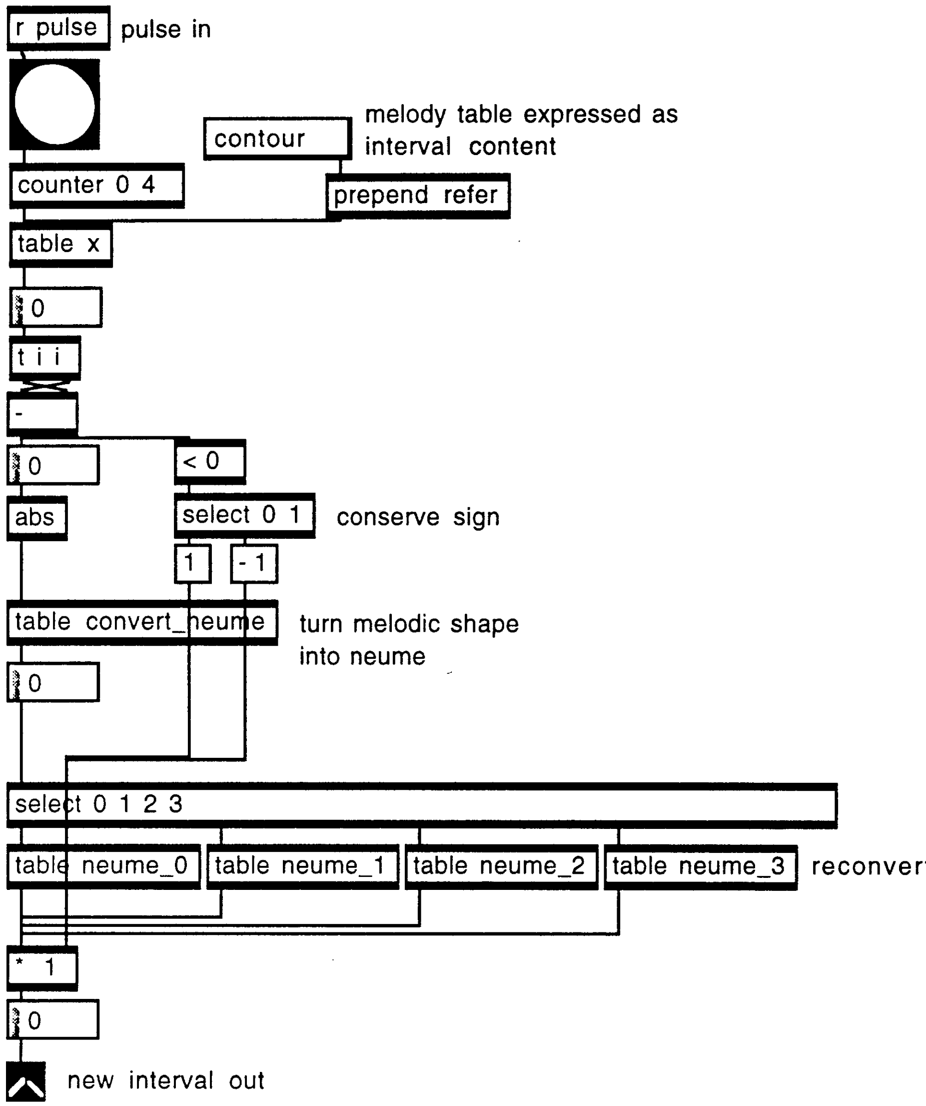
chord_play



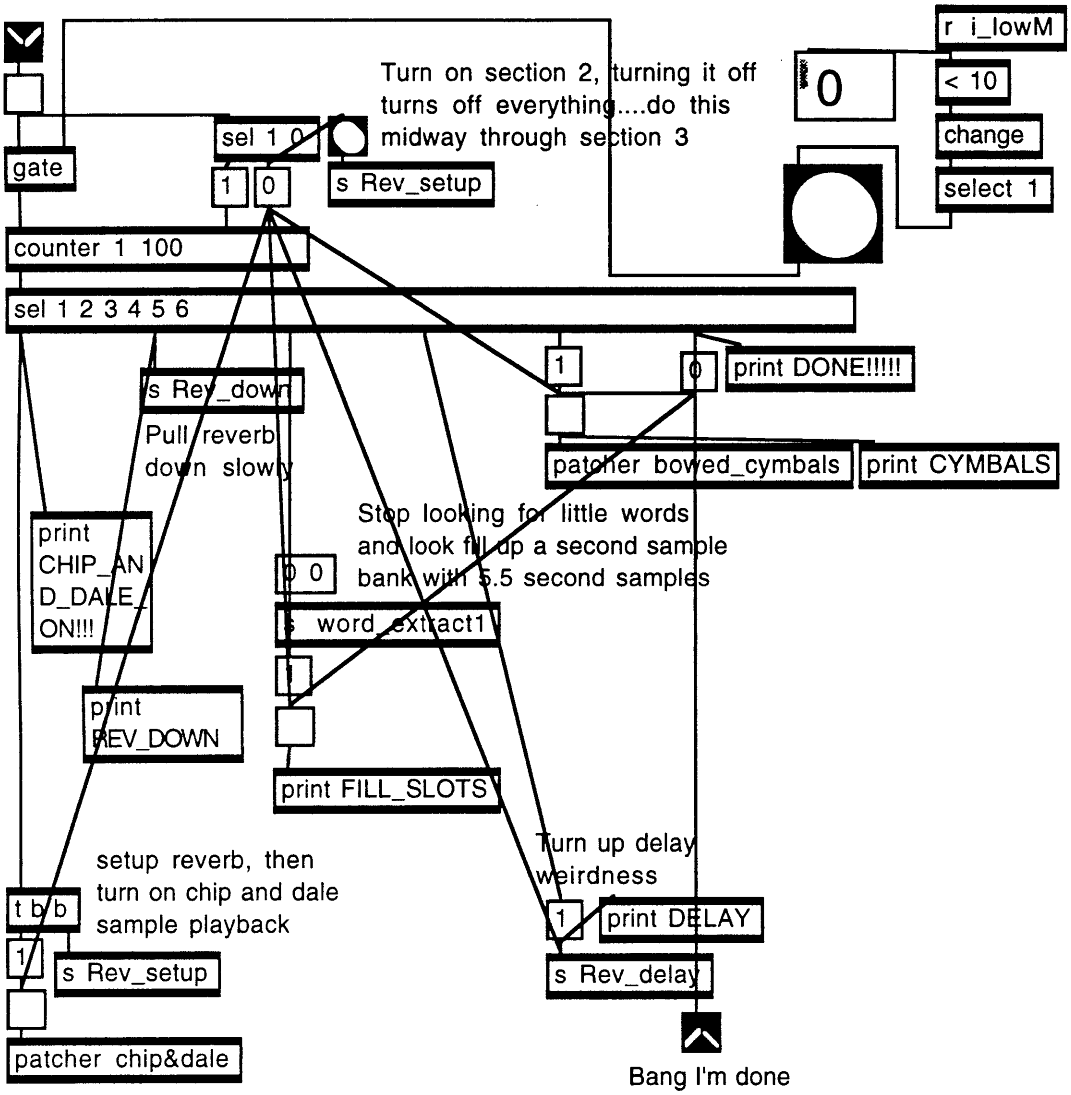
control_sequence



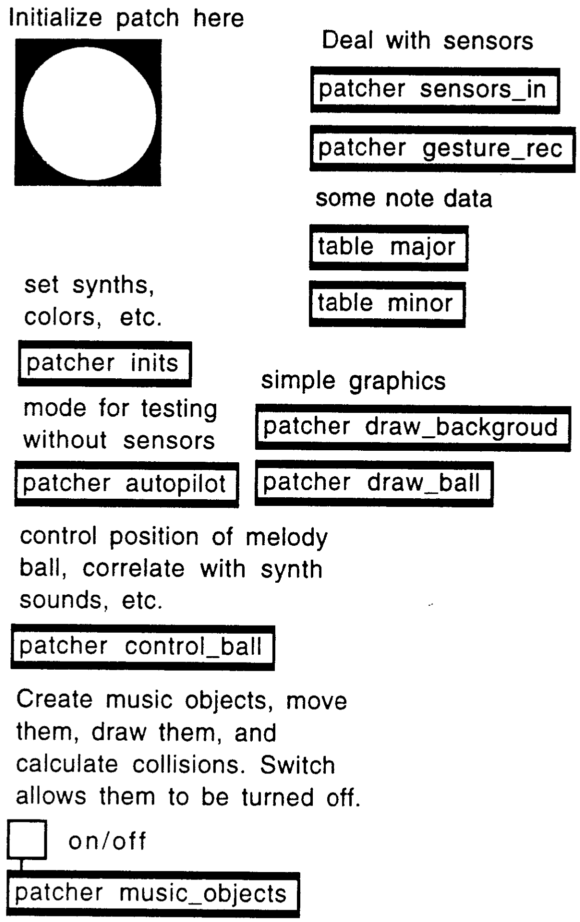
convert_neume



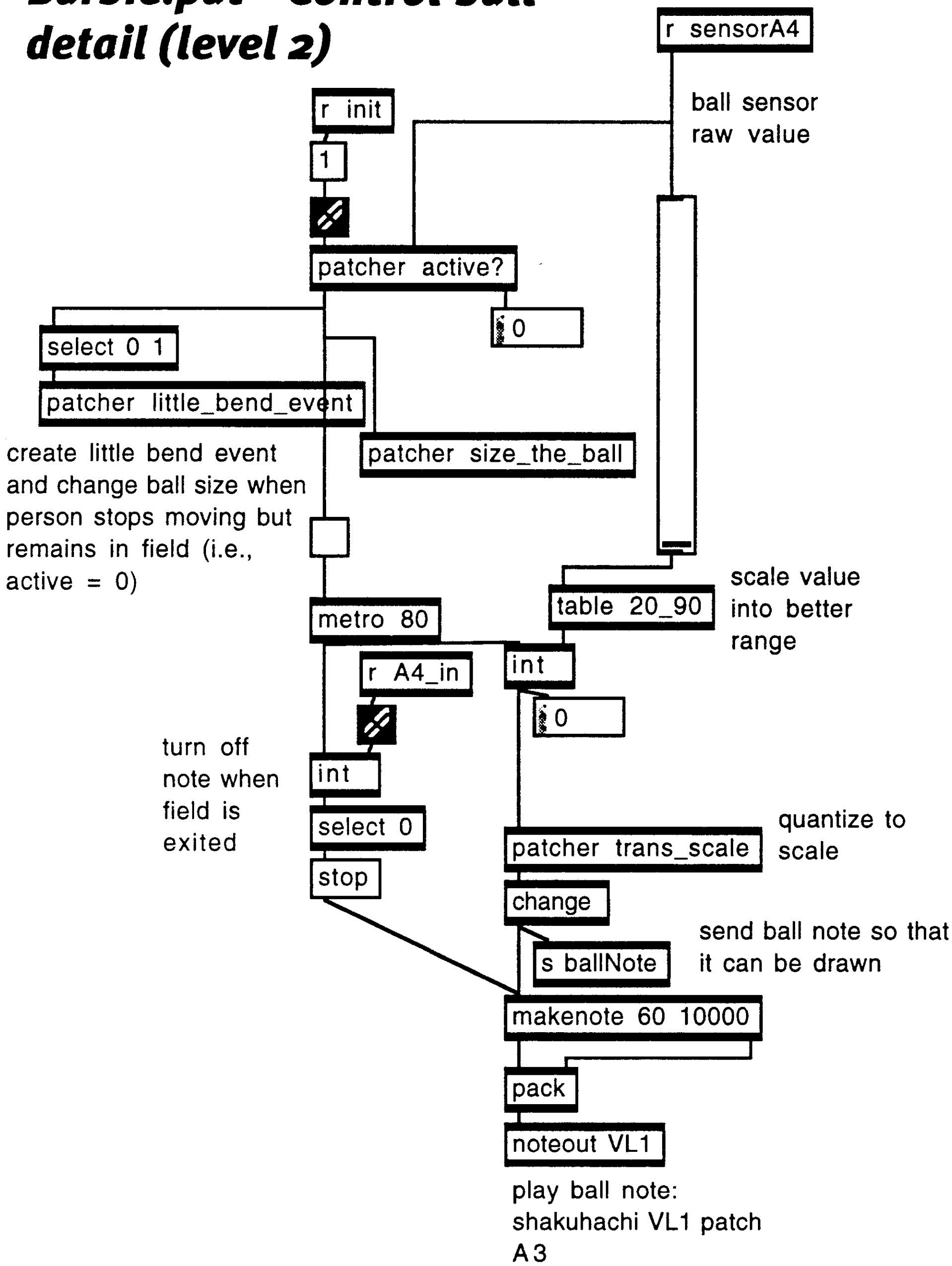
Example section (change_section) from Hand Gesture Music



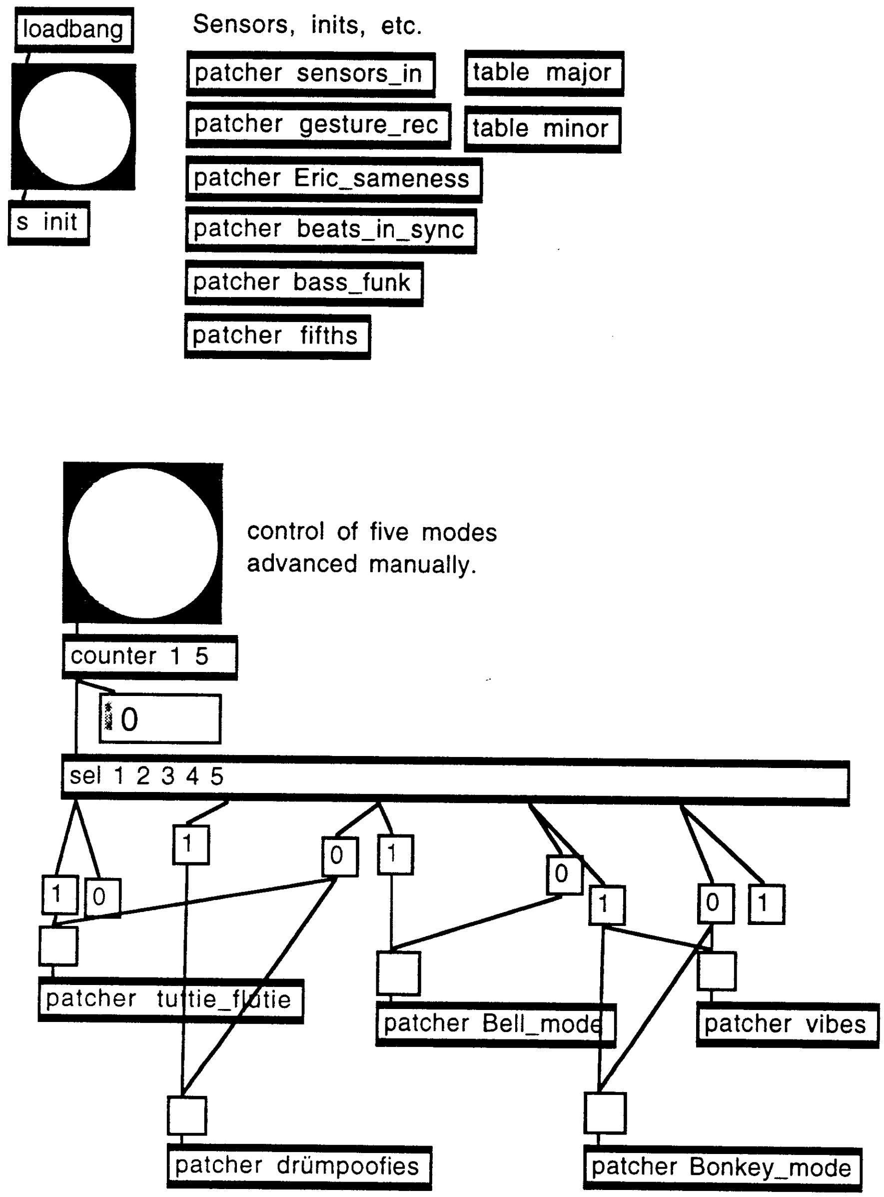
Barbie.pat - Top Level



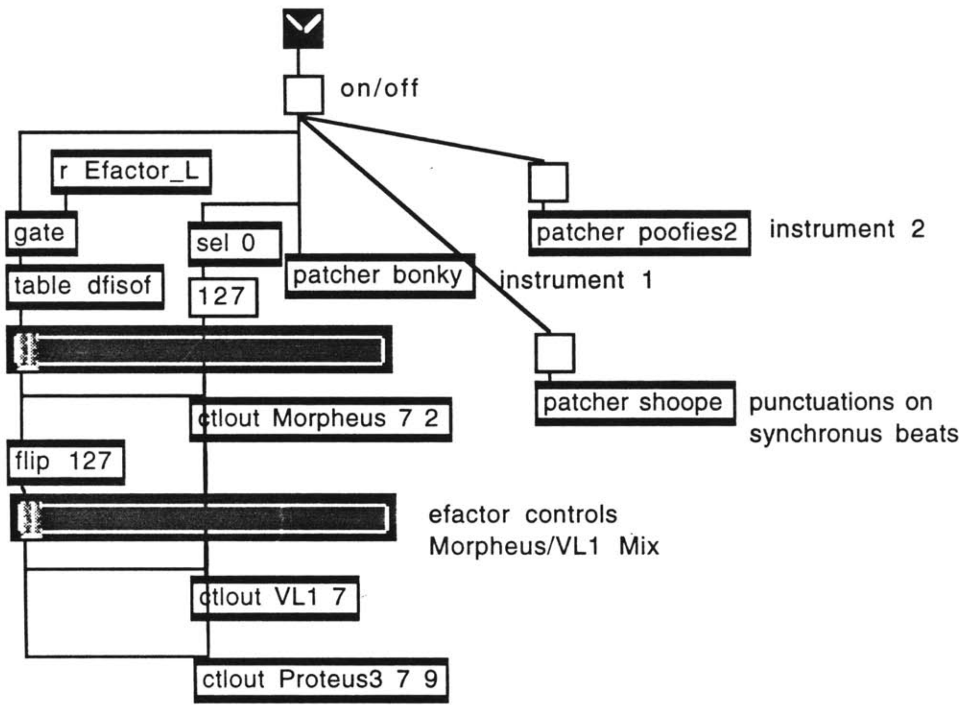
Barbie.pat - Control ball detail (level 2)



The Sensor Frames: Top-Level



The Sensor Frames: Drum Mode detail (level 2)



The Sensor Frames: Drum mode, beat punctuations detail (level 3)