Mapping transparency through metaphor: towards more expressive musical instruments

SIDNEY FELS,† ASHLEY GADD† and AXEL MULDER‡

†Human Communication Technologies Laboratory, Department of Electrical and Computer Engineering, 2356 Main Mall, Vancouver, BC V6T 1Z4, Canada
E-mail: {ssfels, gadd}@ece.ubc.ca
‡Infusion Systems Inc., P.O. Box 16178, North Vancouver, BC B7J 3S9, Canada
E-mail: mulder@infusionsystems.com

We define a two-axis transparency framework that can be used as a predictor of the expressivity of a musical device. One axis is the player's transparency scale, while the other is the audience's transparency scale. Through consideration of both traditional instruments and new technology-driven interfaces, we explore the role that metaphor plays in developing expressive devices. Metaphor depends on a *literature*, which forms the basis for making transparent device mappings. We examine four examples of systems that use metaphor: Iamascope, Sound Sculpting, MetaMuse and Glove-TalkII; and discuss implications on transparency and expressivity. We believe this theory provides a framework for design and evaluation of new human–machine and human–human interactions, including musical instruments.

1. INTRODUCTION

Why is it so difficult to make a novel expressive musical device? This paper provides a framework for understanding and predicting expression of devices and their mappings. We consider *transparency* as a predictor for expressivity. We explore the role of metaphor for improving the amount of expression possible with a device. Metaphor depends on a *literature*, which forms the basis for improving transparency. We discuss four systems, Iamascope (Fels and Mase 1999), MetaMuse (Gadd and Fels 2002a, b), Sound Sculpting (Mulder, Fels and Mase 1999), and Glove-TalkII (Fels and Hinton 1998). Each system's use of metaphor has interesting implications on transparency and expressivity.

In this paper, we will introduce transparency as a quality of a mapping. Similar to Moore's (Moore 1988) notion of control intimacy, transparency provides an indication of the psychophysiological distance, in the minds of the player and the audience, between the input and output of a device mapping. The more transparent the mapping is, the more expressive the device can potentially be. New technologies are often poorly understood, and therefore tend to produce opaque mappings. Metaphor is one technique to facilitate moving from an opaque mapping to a transparent mapping.

Metaphor enables device designers, players and audience to refer to elements that are 'common knowledge' or cultural bases which we call *literature*. By grounding a mapping in the literature, it is made transparent to all parties. Metaphor restricts and defines the mapping of a new device. Through metaphor, transparency increases, making the device more expressive.

We examine four systems that use metaphor and discuss the lessons learned from these systems. First, we consider the Iamascope, an interactive video kaleidoscope that uses metaphor to explain its musical control. Iamascope uses a guitar metaphor to explain the technology-based musical mapping *post hoc* to help participants play music with it. Lack of expression occurs where the metaphor breaks down due to the limited input range of the system. We then consider Sound Sculpting, which uses the metaphor of sculpting clay to change the shape of a virtual object. The shape of the object then affects the parameters of an FM synthesizer. The metaphor works for parameters such as spatialisation, but fails with the less intuitive parameters of FM synthesis.

Third, we consider MetaMuse, a controller for granular synthesis. The prop-based control of MetaMuse is based on the metaphor of rainfall, which matches the process of the synthesis engine. Parts of the mapping are transparent, but MetaMuse also has difficulties, as the discrete nature of sample selection does not fit the metaphor well. Finally, we consider Glove-TalkII, an adaptive gestural controller for formant speech synthesis. Glove-TalkII uses hand gestures that match the movements of the lips and tongue during normal speech. It is unique among these systems in that it adapts to the speaker's understanding of the mental model. The use of metaphor in Glove-TalkII makes the complex gesture set cognitively manageable for the novice speaker.

The framework of expressivity and metaphor is presented in this paper with respect to sound and music devices. It may also be applied to other fields of human interaction, including human–human, human–computer, and human–machine interaction.

2. TRANSPARENCY, EXPRESSIVITY AND LITERATURE

We consider expression to be a communicative act in which the player and the listener are both responsible for

Organised Sound 7(2): 109–126 © 2002 Cambridge University Press. Printed in the United Kingdom. DOI:10.1017/S1355771802002042

determining to what extent a performance is expressive. Expression is the act of communicating meaning or feeling. Both player and listener, therefore, are involved in an understanding of the mapping between the player's actions and the sounds produced. The mapping, and the ease of understanding it, are therefore critical to determining the success of an instrument.

Both player and listener understand device mappings of common acoustic instruments, such as the violin. This understanding allows both participants to make a clear cognitive link between the player's control effort and the sound produced, facilitating the expressivity of the performance.[1] For many instruments, this link is sufficiently integrated into the culture as to make it bi-directional. In this situation, observing either the sound or the effort provides access to the other. For example, one can picture the vigorous sawing of a virtuoso violinist while listening to an audio-only recording of a particularly exuberant performance. Likewise, watching a good pantomime of a vigorously sawing virtuoso violinist evokes an expressive sound performance. Together, the effort and the sound reinforce one another, increasing the expressivity of the performance. Instruments with a strong link between control effort and sound are more likely to become part of the literature.[2]

2.1. Transparency of device mappings

One of the key attributes of instruments required for adoption into the literature is expressivity; this is a necessary condition for acceptance. We argue that the expressivity of an instrument is dependent on the transparency (defined below) of the mapping for both the player and the audience. With this factor in mind, we can attempt to identify how an instrument, based on a new technology, can make its way into the literature and become a referent. This course depends in large part on the mapping from control to sound.

The mapping component is placed within the larger context of the instrument or device in figure 1. The device itself is composed of three parts: the input interface, the mapping, and the output interface. The input interface consists of the set of control gestures used to control the device. This is different from the physical input device, which can restrict or suggest certain control gestures but also interprets them, so has a mapping aspect. The output interface consists of the possible range of sound outputs that the device can make, as distinct from the actual synthesis engine used. The mapping defines how the control gestures translate into sound output and comprises the whole system, from the input interface to the output interface. This is important because understanding the mapping is critical to the expressivity of the device.

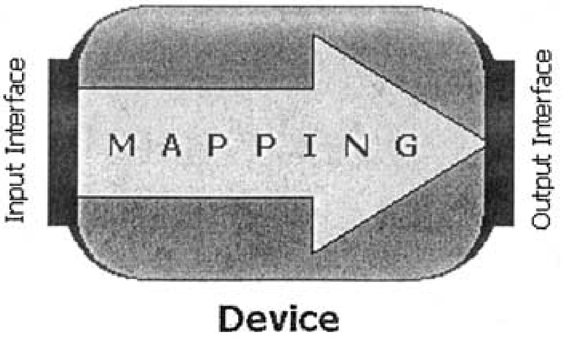


Figure 1. The musical device has an input interface and an output interface. The two are related by the mapping.

In the case of traditional acoustic musical instruments, physics drives the mapping between control and sound. Traditional instruments are typically implemented with mechanical systems. As such, the mapping usually is easily understood by the player. Further, the physical form factor makes learning to play the instrument possible on a reasonable human time scale. These two factors make the mapping between instrument control and sound production psychophysiologically *transparent* for the player. Similarly, the audience's understanding of the instrument benefits from the physical nature of the mapping. The audience also benefits from a long cultural association with traditional instruments, expecting certain inputs to result in certain outputs. Both of these factors make the mapping transparent for the audience. Transparency for both the player and the audience makes expressivity possible.

As an example, the acoustic guitar is a well-known instrument. The lay audience understands the manner in which the player's control gestures map to sound output, even if they lack the physical proficiency to play the guitar themselves. This common understanding makes the guitar's mapping transparent to the audience. With enough practice, it also becomes transparent to the player. Under these (common) conditions, the guitar is an expressive instrument.

The advent of electronic musical instruments complicates the understanding of whether a musical instrument is expressive. This complication arises because such instruments allow the separation of control from sound (Winkler 1995, Hunt, Wanderley and Kirk 2000, Jorda 2001). Most modern synthesis engines are controlled by time-varying sets of numerical parameters. These parameters can be produced in many ways and by using many different mappings. This *physical* separation requires an effort on the part of the designer to avoid the corresponding cognitive separation. Many instruments based on these engines have arbitrary mappings,

[1] Expressivity is not guaranteed – expression is complex, and transparency facilitates expression.

[2] Here we are distinguishing the concept of *literature* from its literal definition of 'that which is written'. What is intended is the more general definition of that body of knowledge understood and accepted as part of a culture. It is 'common knowledge' and is used as referent rather than being explained by reference to something else. For example, scents are often compared to that of a rose, but the scent of a rose is never identified by comparison to something else.

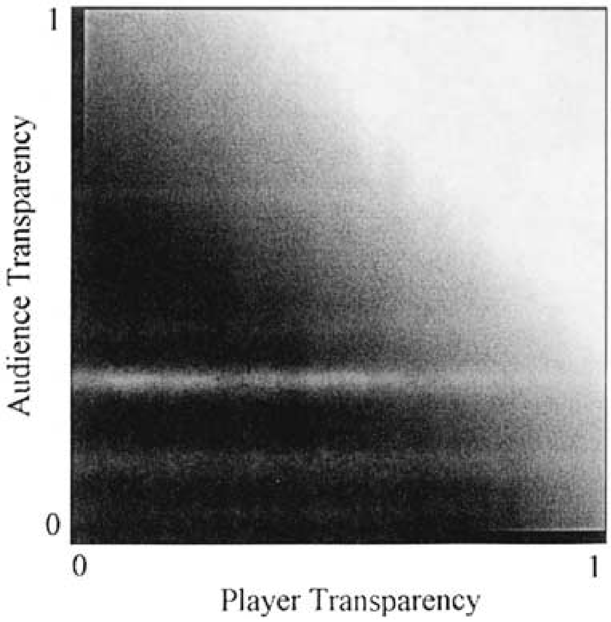


Figure 2. The graph created by mapping transparency for the player and for the audience.

which can make the mapping very opaque to both player and listener. Learning an opaque mapping is difficult for both parties, making expressivity problematic.

The synthesizer keyboard provides an excellent example of how control and sound can become separated. One of the presets for many synthesizer keyboards maps key presses to a variety of percussion sounds. However, the standard mapping, in which pitch increases to the right, is not valid for percussion instruments. This means that the different sounds are mapped somewhat arbitrarily to the keys. While it may be apparent that individual key presses map to individual sounds, the specific mapping is opaque to both the player and the audience. Learning to play percussion on the synthesizer keyboard is very difficult, as is understanding such a performance.

These examples suggest a two-dimensional continuum of mapping transparency, with one axis for the player and one for the audience. The transparency of each axis varies between 0 and 1, as shown in figure 2. The transparency of the mapping depends on different factors for the player and the audience.

The transparency of a mapping for the player depends both on cognitive understanding and on physical proficiency. Cognitive understanding requires that a player must be familiar with the expected effects of the control parameters on the sound output. Such familiarity can be improved by exposure to performances with the instrument. Proficiency is the level of dexterity that a player has with the controls, and therefore can improve with practice. Thus, familiarity and practice make a mapping more transparent for the player. This concept is very similar to Moore's concept of control intimacy:

> The best musical instruments are ones whose control systems exhibit an important quality that I call 'intimacy'. Control intimacy determines the match between the variety of musically desirable sounds produced and the psychophysiological capabilities of a practiced performer. (Moore 1988)

Moore's control intimacy, however, refers to the entire device, whereas transparency refers specifically to the mapping between the input and output interfaces. The player's degree of transparency provides one axis for evaluating and predicting the expressivity of the device.

The audience's degree of transparency provides an orthogonal axis. However, the audience does not require physical proficiency with the interface. Instead, they only need to have an understanding of how the instrument works to appreciate the proficiency of the player. For the lay audience, this understanding is derived from cultural knowledge, including percepts of physical causality relationships, which we have called the literature. Interestingly, this model would predict that it is possible for the *audience* to increase the expressivity of the instrument. This could be accomplished by studying the theory of the instrument or by learning to play the instrument, both of which would increase the transparency of the mapping. Increased transparency contributes to the audience's appreciation of the player's proficiency, leading to increased expressivity.

2.2. A framework for expressivity

We have defined orthogonal axes representing mapping transparency for both the player and the audience. Though the axes are continuous, for referential convenience we roughly divide the square into four quadrants, as shown in figure 3. Then, OT refers to the region that is opaque for the player but transparent for the audience, and so on.

Most traditional instruments lie in the TT quadrant, transparent for both the player and the audience. The violin, for example, is well known to both player and audience due to cultural exposure. The mapping of control gestures to sound output is embodied in the mechanical construction of the instrument. This embodiment, along with the form factor of the instrument, makes the affordances (Norman 1990) of control apparent to the player and the audience. Because the violin is a culturally familiar instrument, the gestures that control it affect the output in known, predictable ways. These gestures include string choice, finger position, and bowing parameters. The violin's form factor and control predictability also make it learnable on a reasonable human time scale, though many young students may complain to the contrary. These attributes make the violin's mapping transparent for both the player and the audience.

On the other end of the spectrum, many new technologies fall in the OO quadrant, opaque for both the player

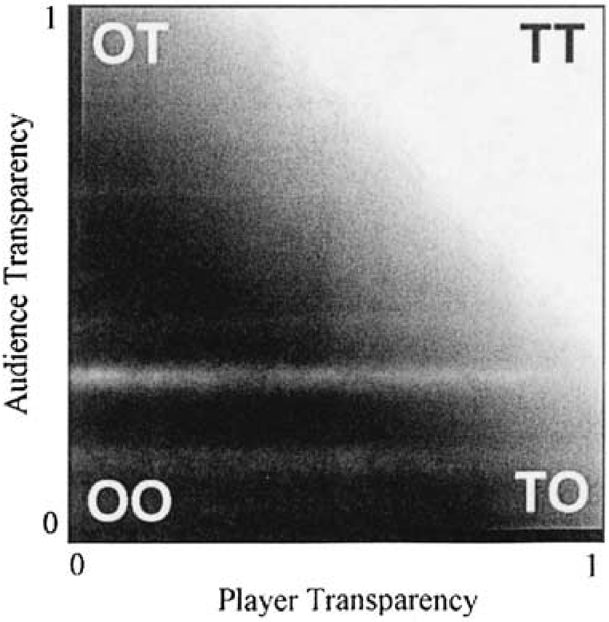


Figure 3. Regions can be identified in the graph. Expressive devices fall in the Transparent–Transparent region.

and the audience. New controllers require both parties to learn the mapping from unfamiliar control gestures to existing output interfaces. New synthesizer engines frequently attempt to create novel sound output spaces, which must be mapped from an existing input interface. The worst-case scenario, new controller mapping to new synthesis engine, is increasingly common. In all these cases, there is a gap in familiarity for both player and audience. Neither party knows what output to expect, based on a given input. The player can improve on this situation by gaining physical proficiency, but this is difficult when the mapping is not clear. The Very Nervous System (VNS) (Rokeby 1995), a gestural controller similar to the Iamascope (described in section 3), is an example of an OO instrument. It uses Fourier analysis to determine the frequency components of the video input, mapping these to musical parameters. The mapping is so complex, however, that it is extremely difficult for either the player or the audience to understand what is happening.

There are two common ways to move a new technology out of the OO quadrant. The first is to make the instrument simple; the second is to add desirable functionality. These methods tend to move instruments in different directions, to OT and TO, respectively. Simplifying an instrument tends to make it easier for the audience to understand, but does not necessarily make it easier to play. Often simplifications reduce the dynamic range of the output, lowering the expressive capacity. Adding functionality creates a motivation for early adopters (Norman 1998) to learn the instrument but provides no explanation of the instrument's mapping to the audience.

The common problem that both of these methods share is that neither of them relates to existing literature. This displacement from a common reference point causes opacity for both player and audience. A new mapping, based on reference to the literature, would avoid such drawbacks. Metaphor can be used to relate new technology to the known, cultural basis of the literature. The literature may be from any culture, and metaphors from two or more literatures can be combined in a device. In the following section, we present metaphor as a way to increase the transparency for both the player and the audience.

2.3. Increasing expressivity using metaphor

The application of a metaphor to an interface has the effect of increasing the transparency for both the player and the audience. However, depending on the mapping type, metaphor is effective through different mechanisms. Depending on whether the mapping is modal or non-modal, or is convergent, one-to-one, or divergent, six possible mapping types exist. Modal mappings are those in which the input is multiplexed temporally. That is, depending on the active mode, a given input can produce one of multiple outputs. Convergent, one-to-one, or divergent mappings are based on the amount of spatial multiplexing – the degree to which groups of gestures are mapped to groups of simultaneous sounds. The six possible mapping combinations are shown in table 1 along with examples.

Modal mappings can benefit from metaphor as a way to obviate the instrument's current mode. Convergent, divergent and one-to-one mappings can all use metaphor to explain their behaviour. The following sections discuss examples of convergent non-modal, convergent modal, and one-to-one non-modal mappings. Finally, metaphor is presented as a design tool.

2.3.1. Convergent non-modal mappings

Convergent non-modal mappings generalise groups of control gestures into common outputs. An example from the literature of musical instruments is the piano. Many finger positions activate the same key, sounding the same note. There are no modes, so the note played is the same each time the key is pressed. Metaphor can be used to cognitively group the control gestures associated with one sound output. In the case of the piano, a range of finger positions is understood to activate a single key. This metaphor has been used in instruments that use a key model but do not have explicit keyboards, such as in the Virtual Piano created by Leonella Taraballa and Graziano Bertini at the CNUCE in Pisa in 1997. The Virtual Piano removes the keyboard entirely, relying on the familiar gestures of a pianist without the physical keys.

Table 1. Combinations of mapping strategies. Metaphor is applied to each combination differently; we discuss the synthesizer keyboard, the virtual piano, and BoSSA in the sections below. Convergent and divergent mappings are also referred to as many-to-one and one-to-many mappings, respectively, in Hunt *et al.* (2000). (*2Hearts* [McCaig and Fels 2002] uses two heartbeats to control multiple parameters such as sequencing, filters and effects.)

		Spatial Multiplexing		
		Convergent	One-to-One	Divergent
Temporal Multiplexing	Non-Modal	Piano, Virtual Piano	Violin, BoSSA	2Hearts
	Modal	Synthesiser Keyboard	Electronic violin	Scratch turntable

2.3.2. Convergent modal mappings

Modal mappings use internal modes to choose which sound output will result from each single gesture. For example, the synthesizer keyboard uses different modes to map convergent key presses to different outputs. Pressing the same key in the same way can, in different modes, produce the sound of a piano, a tuba, a raindrop, or any other arbitrary sound. In this case, the piano keyboard metaphor, which has pitch increasing to the right, can be maintained if the sounds produced contain a pitch element. However, the mode selection is arbitrary, hidden from the audience. Furthermore, it is often poorly indicated to the player, usually consisting of a set of buttons with some indicator light, or a menu system. This interface could be improved with the application of an appropriate metaphor defining and explaining the mode selection process. One rather simplistic solution would be to use a tangible interface (Ishii and Ullmer 1997) based on small figurines of actual instruments. These would be placed on the keyboard to indicate mode selection to the player and the audience. The obvious problem with this metaphor is that it requires the player to find the correct figurine in order to switch modes during a performance. This may be too time-consuming, especially in instruments with many tens or hundreds of possible modes.

2.3.3. One-to-one non-modal mappings

One-to-one mappings exemplify a direct relationship between control and output. With a complex instrument, it can be difficult to remember what the relationship is. Metaphor can be used to provide a control framework for the mapping. This framework creates relationships to the individual control gestures. BoSSA (Trueman and Cook 1999, Bahn and Trueman 2001), for example, bases its control gestures on those of the violin. Instead of directly affecting a vibrating string, the BoSSA player bows a set of force sensing resistor-based vanes, while fingering a pressure-sensitive fingerboard on an attached neck. In this way he directly interprets the violin metaphor. BoSSA then builds on that base by allowing gestures not normally useful on the original instrument, such as changing the angle of the neck relative to the body of the instrument.

One interesting offshoot of this approach is the possibility of combined mapping types. The acoustic guitar, for example, is similar to a violin in its control gestures. However, it also incorporates components of a convergent mapping through the inclusion of frets. Frets allow many finger positions on the strings to be mapped to one string length, which produces a single sound output. The use of frets improves the transparency of the instrument by making it more apparent which finger positions will produce which notes. Novice violinists spend a long time learning the correct finger positions for each note, while frets ease this process for novice guitarists. This increase in transparency comes at the expense of expressivity.[3] Guitarists can no longer create glissandos, trills or vibratos using the same gestures as violinists. However, guitarists have found ways to regain this expressivity that would not be possible without the frets. Pitch bends are accomplished on a guitar by sliding the string sideways on the fret, thereby stretching the string. Vibrato can also be achieved by varying finger pressure behind the fret, also stretching the string. Such gestures are not possible on a violin because they require frets, and because the cocked wrist position of a violinist does not provide a strong enough grip to affect the strings in these ways.

As an aside, one variation for the guitar, suggested by this comparison to the violin, would be to remove the frets after the player has learned the correct note positions. In this case, the frets would act as training wheels

[3] This demonstrates the idea that transparency is a necessary but not sufficient condition for expression. In this case, increasing transparency has decreased the dynamic range of the instrument, which decreases its expressivity.

for the guitarist. Removed when no longer needed, the guitarist could then return to the more transparent one-to-one mapping of a fretless guitar. Indeed, there is a growing community devoted to the subculture of fretless guitar.

2.3.4. Metaphor as a design tool

We have seen that metaphor can be applied to new technologies in many ways in the previous sections and in Marx (1994) and Svanæs and Verplank (2000). Metaphor can also be used as a design tool when creating new instruments. If a new synthesis engine is implemented, it may suggest a metaphor that encompasses its main characteristics. The metaphor may then dictate an appropriate controller for the device, so that the entire device is self-consistent. For example, MetaMuse, presented in section 5, is based on granular synthesis. The discrete event-based nature of granular synthesis suggested the rainfall metaphor used in the device, which then indicated a watering can as an appropriate controller. This design strategy can also be reversed: a new controller may suggest a metaphor, which may then dictate an appropriate synthesis engine for the device. Finally, an instrument can be based on an original metaphor, from which both the input and the output interfaces are drawn. By applying these design strategies to the mapping types discussed above, metaphor can lead to more transparent instrument mappings, which in turn create expressive devices.

The authors have used metaphor in four systems in past research: Iamascope, Sound Sculpting, MetaMuse and Glove-TalkII. In subsequent sections, we will retrospectively examine how these systems use metaphor to make them more expressive. We will see that these systems are consistent with our theory, both in their benefits and in their shortcomings.

3. IAMASCOPE: A METAPHOR FOR A VIDEO CONTROLLER

The Iamascope is an interactive kaleidoscope that uses computer video and graphics technology. In the Iamascope, the performer becomes the object inside the kaleidoscope and sees the kaleidoscopic image on a large screen in real time. The Iamascope is also a music controller. This functionality was added to allow the participant to play music at the same time as they play imagery. Originally, the musical control was technology driven, but proved difficult for participants to understand how to play. So, without changing the mapping, we created a metaphor based on a guitar to help people understand it. This is an interesting use of metaphor to increase transparency without changing the mapping.

A block diagram of the Iamascope is shown in figure 4. For input, the Iamascope uses a single video camera whose output is distributed to two separate video processes: one for imagery and one for sound. Imagery output from the Iamascope is displayed on a wall-sized projection screen. Audio output from the Iamascope is played though stereo speakers beside the display. In the current implementation, a pie slice from the video image is selected to form the original image (O), which is used to create the desired reflections (O') for the kaleidoscope. The image processing part of the vision-to-music subsystem uses the exact same pie slice (O) for the music. In this way, movements that cause kaleidoscope effects cause musical effects. A picture of a person using the Iamascope is shown in figure 5.

The kaleidoscope subsystem maps the participant's movements to imagery in a direct, one-to-one manner. This mapping is discussed in Fels and Mase (1999). Of interest here is the gesture-to-music mapping. The musical mapping maps active *zones* to musical notes as discussed in the following section. The feedback available to the participant comes from sound, video and proprioception.

3.1. Vision-to-music subsystem

The vision-to-music subsystem has two parts, image processing and music production. The image processing is responsible for capturing the video image, extracting the correct part of the image and calculating intensity differences. The music production part is responsible for converting a vector of intensity differences into MIDI signals to control a MIDI synthesizer.

3.1.1. Image processing

A block diagram of the image processing system is shown in figure 6. The function of the image processor is to divide the active video region into bins and compute the average intensity difference between the current bin and the previous bin (in time). Normally, ten bins are used. The vector of intensity differences for all the bins is sent to the music production part of the subsystem. All the image processing code is written in C.

3.1.2. Music production

The music production part of the vision-to-music subsystem runs every time a new vector of bin intensity differences is received from the image processor. Many schemes are possible for musical control based on the input from the image processor. A production scheme that did not require any absolute positioning of the body and plays euphonic music to match the beautiful kaleidoscope images was chosen. Within these constraints, there is room for some musical control and expression by the performer.

In the current system, the musical key is selected by the computer. Each bin represents a semitone offset from the root note of the current key. The offsets are chosen so that each bin in ascending order is associated with a I, III or V note from the current key in ascending order,

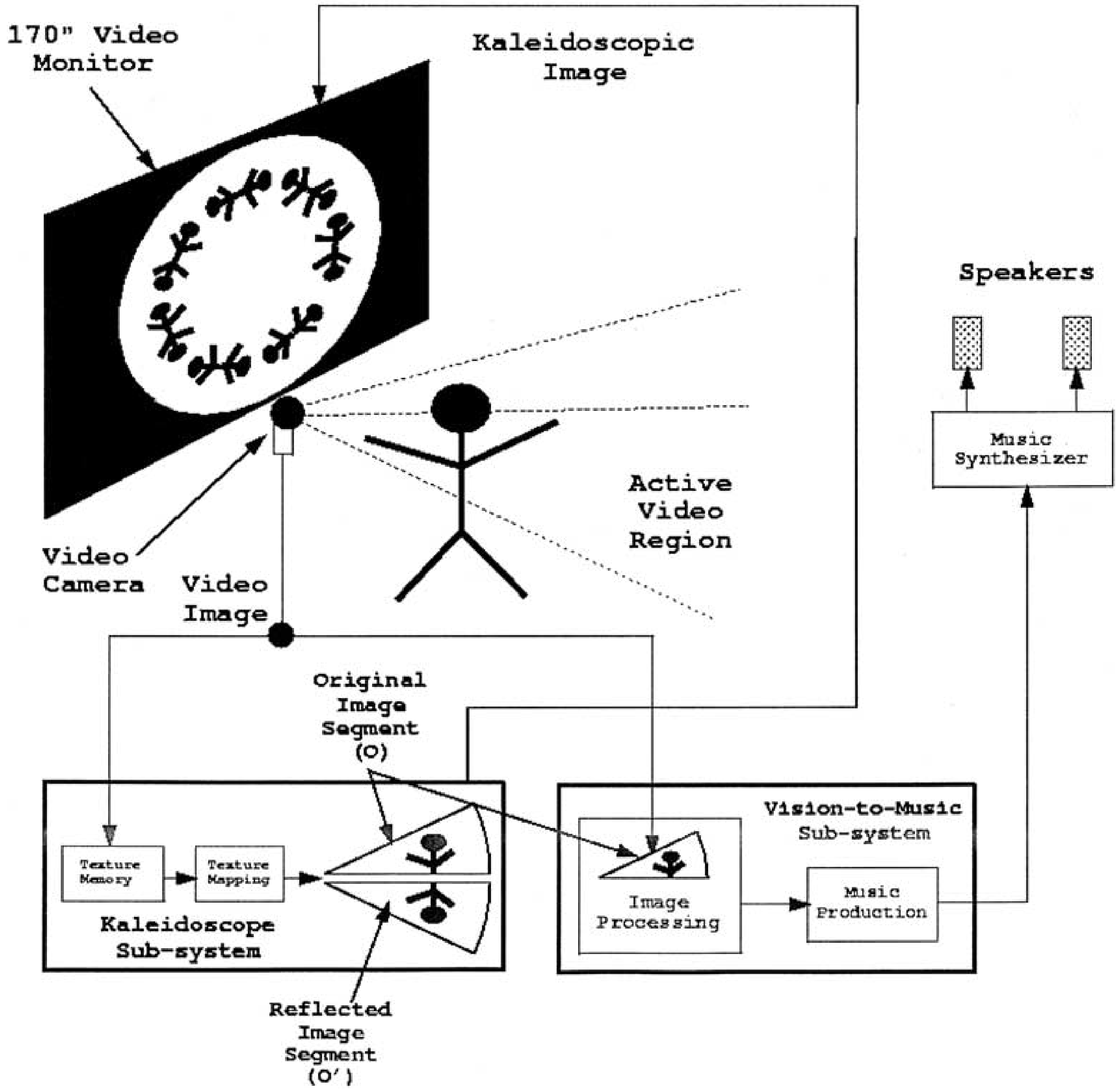


Figure 4. Block diagram of the Iamascope. Output from the video camera feeds into both the kaleidoscope subsystem and the vision-to-music subsystem.

providing consistently harmonic sounds. For example, if the current key is C then bin 0 represents a 0 offset (C note), bin 1 represents an offset of 4 (E note), bin 2 represents an offset of 7 (G note), bin 3 represents an offset of 12 (C note, one octave higher), and so on. A note plays when the image intensity difference for a bin exceeds a threshold. The note velocity is controlled by the intensity difference. Notes turn off as a function of time and intensity change as described in Fels and Mase (1999).

3.2. Mapping and expression

The musical mapping in the Iamascope is mostly technology driven. The algorithm uses a simple video processing technique to map a player's movements to MIDI notes. The player's movements are unconstrained and the player has to discover the mapping on his own. The closest metaphor is that the interface is like a ten-string guitar where the computer holds down the chords automatically. The player strums the strings by moving in the bins. While this metaphor helps make the mapping easier to understand it does not help in learning to play the device. This is because the metaphor is not quite accurate. The Iamascope's musical mapping suffers from two shortcomings:

(1) Players do not know where the strings are since they cannot see or feel them. This makes note timings very difficult and thus the music lacks expression; this is a technological shortcoming as haptic feedback could restore the metaphor.

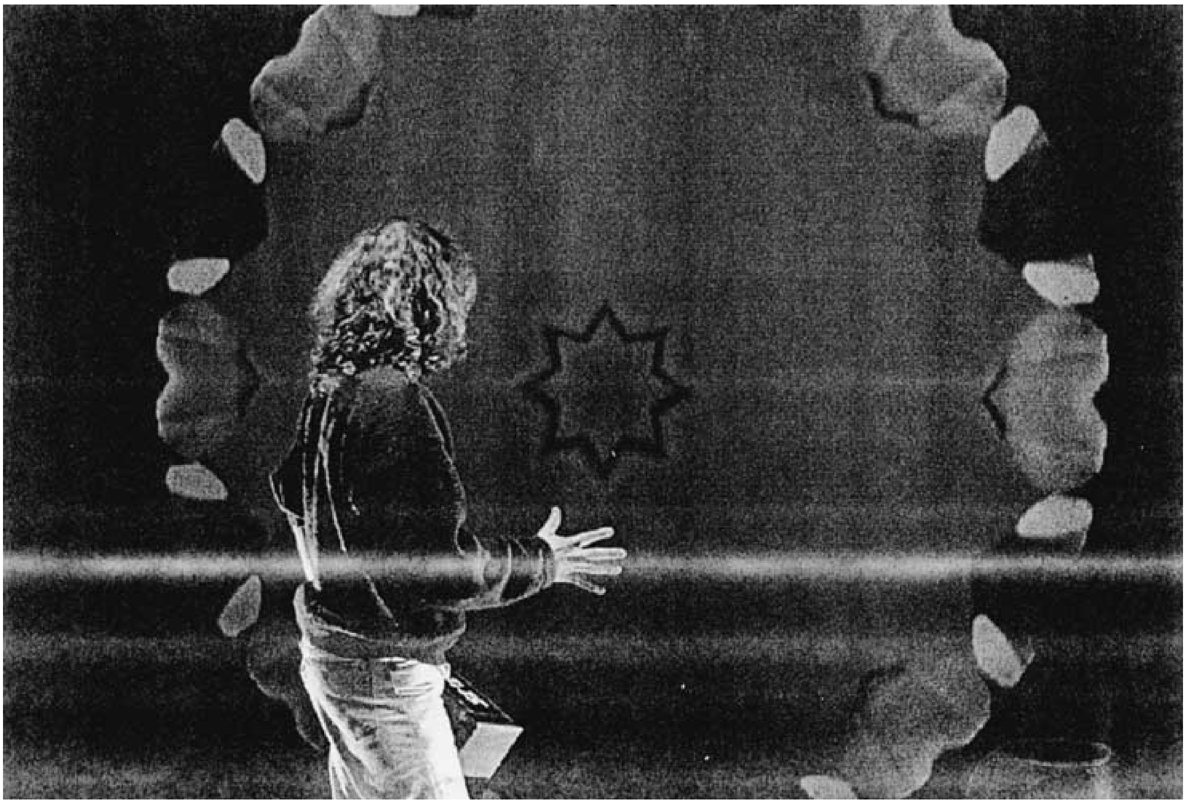


Figure 5. Example of a person enjoying the Iamascope.

(2) Players cannot select their own chords, restricting expressivity. This is a mismatch of the strict guitar metaphor. A different approach may solve this problem.

In general, this attribute of free hand or free form gesture mapped to sound is problematic. Very few metaphors provide a strong enough link between gesture and output to provide an easy-to-learn mapping. Thus, even if the metaphor and mapping are easy to understand, they will not necessarily lead to a very expressive instrument. In this situation, other paths to achieve transparency need to come into play to make the instrument expressive, as discussed in section 2. One metaphor that we explored that does provide a strong link between gesture and effect is the hand manipulation of non-rigid objects such as balloons and rubber sheets. We explored this tight coupling for a metaphor in Sound Sculpting.

4. SOUND SCULPTING: A METAPHOR FOR SOUND DESIGN

Sound Sculpting (Mulder *et al.* 1999) is a controller for sound design, which involves navigation through the multidimensional parameter space of a synthesis engine. It uses the metaphor of sound embodied in a small object. Manipulations of the object produce corresponding manipulations in the sound output.

The goal of a sound designer is to find the correct set of parameters to produce a specific sound. Common controllers for this task centre on the keyboard and mouse. These input devices, however, are not well suited to smooth navigation through high dimensional spaces. One controller that may be better suited to this task is a glove-input device, which permits the hand, through gesture, to simultaneously vary many (possibly correlated) parameters with ease.

Previous work in the use of gesture as a controller has mainly centred on formal gesture recognition. It has been noted (in Fels and Hinton (1993), for example) that, since humans do not reproduce their gestures very precisely, natural gesture recognition is rarely sufficiently accurate. Classification errors and segmentation ambiguity cause many of the problems with gesture recognition. Only when gestures are produced according to a well-defined formalism, such as in sign language, does automatic recognition have acceptable precision and accuracy (Kramer and Leifer 1989). However, the use of a gesture formalism requires tedious learning by the player. Free gestures in unconstrained space, however, are difficult to control.

Metaphor allows the player to hold a mental model of the gesture space. The mental model constrains gestures to a meaningful space if it is sufficiently strong. Using pseudo-haptic feedback with isometric input devices by Lecuyer, Coquillart, Kheddar, Richard and Coiffet (2000), for example, creates a compelling physical sensation using virtual haptic feedback.

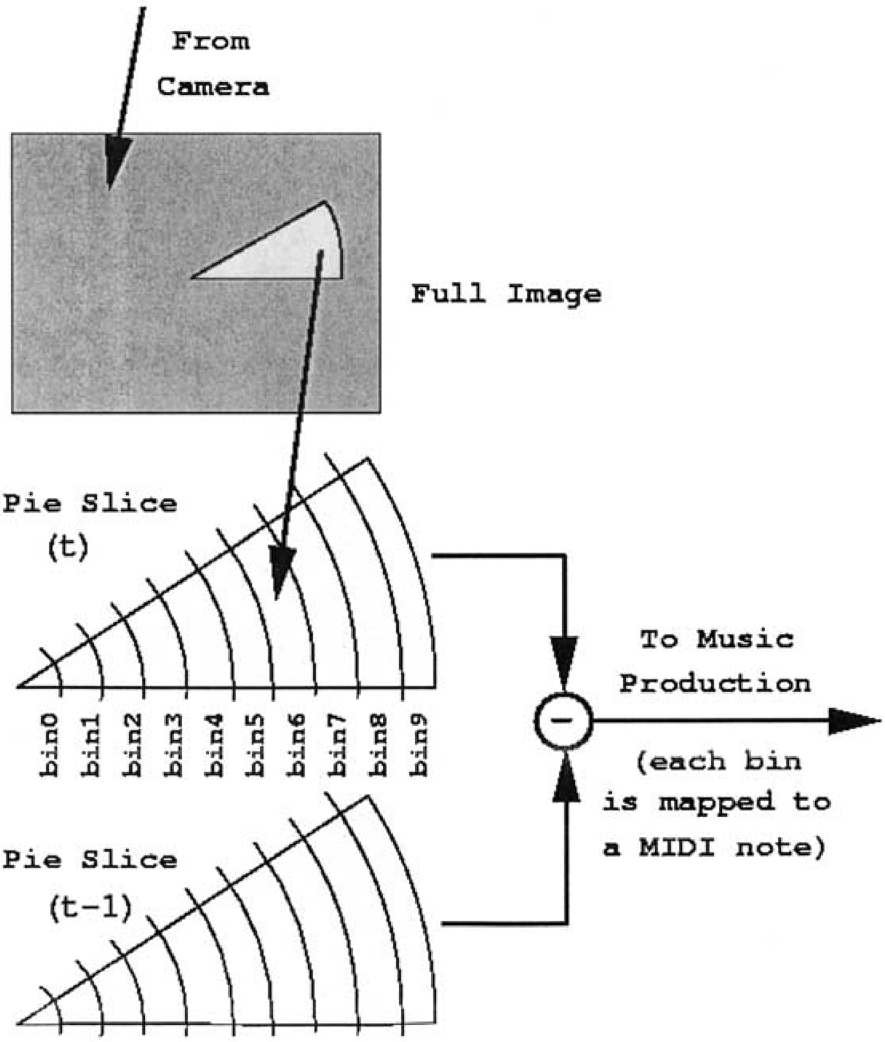


Figure 6. Diagram showing image processing in the vision-to-music subsystem.

In Sound Sculpting, a virtual object is used an as input device for the editing of sound. The sound artist literally 'sculpts' sounds using a virtual sculpting computer interface (Galyean 1991), i.e. by changing virtual object parameters such as shape, position and orientation. The mapping was designed based on pragmatics, and can be explained using the metaphor of sound embodiment.

4.1. Pragmatic-based design

Sound Sculpting applies pragmatics to the metaphor of small object manipulation. We consider object manipulations such as changing the position, orientation and shape of an object. The pragmatics for position and orientation manipulations on small, light objects are simple and do not involve any tools. An analysis of the methods employed by humans to edit shape with their hands leads to the identification of four different stereotypical methods. The methods are:

(1) *Claying*. The shape of objects made of material with low stiffness, like clay, is often changed by placing the object on a supporting surface and applying forces with the fingers of both hands.
(2) *Carving*. The shape of objects made of material with medium stiffness, like many wood materials, is often changed by holding the object in one hand and applying forces to the object using a tool like a knife or a file.
(3) *Chiselling*. The shape of objects made of material with high stiffness, like many stone materials, is often changed by placing the object on a supporting surface and applying forces to the object using tools like a chisel held in one hand and a hammer held in the other.
(4) *Assembly*. Using pre-shaped components, a new shape is created or an existing shape is modified. One hand may be used for holding the object, while the other hand places a pre-shape component.

Sound Sculpting uses the pragmatic of claying to define its gesture set.

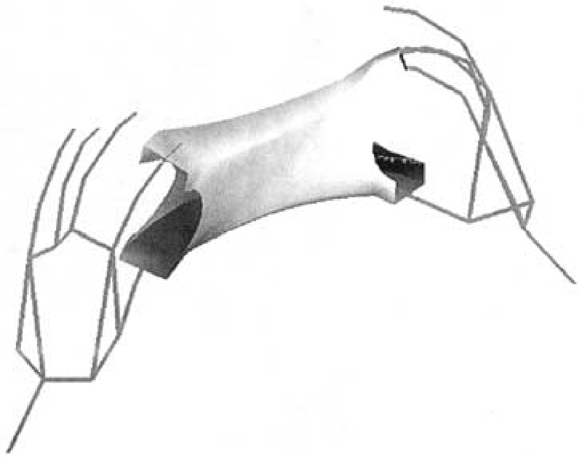


Figure 7. Example of the sheet clamped to the index and thumb tips of both hands.

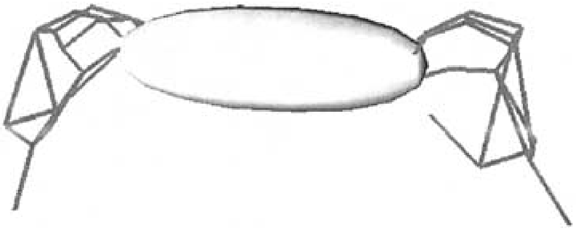


Figure 8. Example of the balloon clamped to both hands.

4.2. Sculpting FM synthesis

Two virtual objects were created and compared for controlling the parameters of FM synthesis: a sheet and a balloon. The claying method used to sculpt these objects was difficult to control without tactile feedback. A derivative method, based more on elasticity, was developed.

A thick rectangular sheet and an elliptical balloon can be virtually manipulated in Sound Sculpting, as shown in figures 7 and 8. Sound parameters such as panning and reverberation are mapped to the virtual positions of these objects. Other FM synthesis parameters, such as flange amplitude, chorus depth, and modulation index, are mapped to object shape properties like length, width and curvature. Pitch and duration of notes were difficult to map to free gestures, so they were either fixed, preprogrammed in a MIDI sequence, or input in real time using a MIDI keyboard.

Manipulation was originally based on touching. The player would reach out with her hand, sensed by a Polhemus Fastrak[4] and a Virtual Technologies CyberGlove,[5] and sculpt the object in virtual space. Although sculpting in the physical world is most effective with touch and force feedback, our assumption was that the metaphor would improve transparency so that haptic feedback would not be necessary. Visual feedback was available, but the intent was that the player would use it solely as a learning tool. Eventually, the player would learn the relationship between shape manipulation and sound output based on the strength of the metaphor, and would not require haptic or visual feedback. This assumption was found to be partially valid. While the player could see and hear the changes made by her actions, it was very difficult to predict where the object actually was. This made motions such as gentle surface strokes difficult.

The claying pragmatic was extended to allow the player to attach her fingertips to control points on the virtual object. This created a more elastic feel to the interface; the player could stretch and pull the object like taffy. This interaction paradigm helped compensate for the lack of tactile feedback.

4.3. Sound sculpting evaluation

Sound Sculpting was evaluated informally, with testing by the author and fifteen research colleagues. Two main conclusions were made.

(1) *Manipulation.* The control of virtual object shape often required some effort to master due to the need for exaggerated movements and/or the need to learn limitations to the control of shape. Due to these limitations to manipulation, unwanted co-articulation of virtual object features could occur. While it is possible that such co-articulation can be used to the performer's advantage in certain tasks (Hunt, Wanderley and Paradis 2002), in the real world the virtual object features used can be controlled separately. The 'touching' of virtual objects was difficult due to a lack of tactile and force feedback, or suitable depth clues.

(2) *Sonification.* The mapping of position and orientation to spatialisation parameters proved easy to use. The mapping of virtual object shape to a variety of timbral parameters offered no obvious analogy to the physical world to the player. Thus, learning was required to obtain desired acoustic feedback in a natural way using the manipulation methods. Forced co-articulation of some shape features prohibited independent control of the sound parameters to which they were mapped. Scaling and offsets of virtual object features for mapping to sound parameters was somewhat arbitrary.

4.4. Sound sculpting: lessons learned

The results of the Sound Sculpting project support our discussion on transparency and the use of metaphor. Parts of the mapping were easily explained, while other parts were obfuscated by the metaphor. Also, one manipulation metaphor was found to be more useful, indicating that the choice of metaphor is important.

[4] A magnetic tracking device.

[5] A dataglove that senses hand posture.

The metaphor of sound embodied in an object worked well for spatialisation parameters such as panning and reverberation. It broke down when the parameters of the sound did not match those of the object. For example, the modulation index of an FM synthesizer does not intuitively map to the qualities of a physical object. A more appropriate metaphor may be useful to control FM synthesis.

Claying and stretching were both implemented in Sound Sculpting. Claying is a compelling metaphor for shape manipulation, but is not useful without tactile feedback. Stretching, however, allows the player's frame of reference to remain attached to the object. The lack of tactile feedback is circumvented at the expense of the ability to vary contact position. This result indicates that it is important to choose a metaphor that can be supported by the input and output interfaces. Claying should be revisited if free-hand tactile feedback becomes technically feasible.

Sound Sculpting uses virtual objects in its metaphor. The next section presents a system that uses real-world objects as props to develop a controller for granular synthesis.

5. METAMUSE: A METAPHOR FOR GRANULAR SYNTHESIS

MetaMuse is a new controller for granular synthesis. Granular synthesis, described by Truax (Truax 1988), blends short, overlapping sound samples to create a gestalt sound, which can be quite different from the original samples. Our controller is based on the metaphor of rainfall.

Current controllers for granular synthesis abstract away the details of the synthesis engine. Specifically, the initiation of each granule is controlled by high-level statistical parameters such as average number of granules per second. The player and audience have no understanding of the process underlying the sound creation, creating opacity in the mappings of such devices.

The process of granular synthesis is very similar to that of natural sound creation. Many natural sounds consist of small, discrete events contributing to the overall sound. Rainfall, for example, consists of the individual sounds of water drops hitting the ground. This process similarity implies that an appropriate controller for granular synthesis could be based on the principles of a sound-producing natural process such as rainfall.

We developed a metaphor based on falling rain. Most people know the sound that rain makes on different surfaces. Using rainfall as a metaphor is seen as a good idea because rainfall is part of the literature. Hence, the metaphor provides a cognitively transparent mapping.

5.1. Design of a particle-driven instrument

We designed and implemented a system that follows the rainfall metaphor as a mapping appropriate for granular synthesis. Props and virtual water are used to support the metaphor of a person controlling the process of rainfall. Props are used to create a source and a sink for the water drops. Props have been shown to be an effective mechanism for interacting with computational models for real-world phenomena (Hinckley, Pausch, Goble and Kassell 1994). Thus it is appropriate to use them for input representations for metaphors. The virtual water falls under a simple gravity model when the source is activated. If it intersects the sink, granules are initiated in the synthesis engine. The props are used to control the parameters of the falling water.

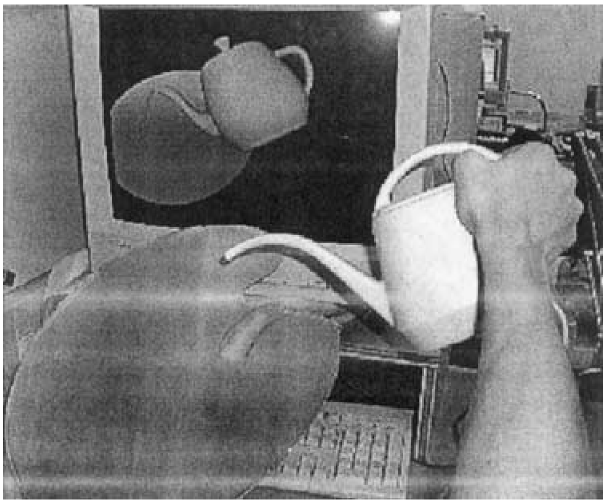


Figure 9. MetaMuse is controlled by two props: a watering can and a palette.

Two props are used in MetaMuse: a watering can and a flat palette, as shown in figure 9. The watering can is the source of the virtual water. It affords the creation of water drops through the motion of pouring. The palette is a sink for the virtual water. It creates a surface on which the drops can land. The drops behave like real rain, falling from the can and hitting the surface.

MetaMuse is played by pouring virtual water from the watering can onto the palette surface. This is done by tilting the watering can while holding the palette below. Both player and audience can imagine the arc of water sprinkling from the watering can and intersecting the palette. This can be visualised using computer graphics, but the strength of the metaphor makes it unnecessary. Many parameters, such as relative height of the props and the position of the drop on the landscape, can be controlled, and the metaphor determines the types of sounds that should be heard. This is easily understood by the player and the audience because it is part of the literature.

The rainfall metaphor is highly appropriate for most aspects of the control. Raising the can higher above the palette will result in a greater impact velocity, increasing the volume and sharpening the sound. Increasing the tilt of the watering can will increase the flow of virtual water, increasing the number of concurrent drop strikes and therefore increasing the number of granules. The rainfall metaphor breaks down when applied to the position of the water drop on the landscape. The metaphor of varying surface composition applies to this mapping, and moving across the landscape should cause a continuous change in the sample played. However, this is technologically infeasible, as the samples are not parameterised. Being pre-recorded, samples are required to change discretely, which does not correspond to the continuous nature of the surface. Therefore the mapping of water drop position to sample is opaque. We are currently investigating techniques to synthesise raindrops with parametric models (Cook 2002).

5.2. MetaMuse implementation

MetaMuse is implemented in C and jMax (IRCAM 2002), with a calibration GUI in Tcl/Tk (Ousterhout 1994). The physical simulation of the water drops is implemented in C and uses a simple physics model. Polhemus Fastrak sensors are mounted on the props to provide position and orientation information to the model through a serial port library. The model is updated in real time, and is visualised using the OpenGL libraries. The visualisation is implemented to assist in debugging and calibration, and is also used to familiarise novice players with the physical model of the system. It is not required for experienced players as the metaphor provides an understanding of how the water flows from the watering can.

There are several controllable parameters in the synthesis engine. The choice of sample, sample rate, and sample volume can all be controlled. Post-processing is also possible, but is not implemented in this version of MetaMuse. The ways in which the parameters are mapped to the controller are dictated by the metaphor.

Droplets are produced at a rate that depends on the tilt angle of the watering can and have an appropriate initial velocity. They then fall freely due to gravity until either they intersect the surface or time out beyond the player’s view. When a drop intersects the surface, its relative position and velocity are calculated and sent to jMax through a UDP connection, initiating playback of a granule. The six parameters of position and velocity are used to calculate the synthesis parameters, which are distinct for each granule.

5.3. Analysis and results

MetaMuse has been implemented as described above. Though no formal user testing has been completed, informal evaluation has illustrated some advantages and disadvantages of the system. Several people of varying backgrounds have played the device, including human–computer interface researchers, musicians, and non-technical non-musicians. Subjects provided feedback on their experiences. Audience feedback was not a priority at this stage of the research, so only a little was gathered.

Subjects reported that the metaphor of falling water is very intuitive and aids in the understanding of the granular synthesis process. This aspect of the mapping is shown to be transparent. However, the metaphor breaks down when players try to vary the position on the landscape. The output does not vary as expected when players pour water onto the different areas of the landscape. This indicates that the implementation of this component of the mapping is insufficient.

This shortcoming is understandable and, in retrospect, could have been predicted. The range of control gestures that vary position on the landscape is continuous. However, the selection process for the granules is discrete; it simply chooses between three different source samples. The mixing of these three samples in the intermediate regions is an insufficient interpolation method, and the resulting sound is not what the player expects. It would be preferable to be able to select from a continuous range of samples, but this is not technically possible. It may be possible to create the appearance of a continuous range of samples by using post-processing or by synthesising the samples in real time with some other synthesis technique.

This deficiency demonstrates a shortcoming of metaphors. The player (and the audience) expects the system to adhere to the metaphor very strictly. When the system deviates, it can cause greater opacity than a system with no metaphor at all. This is because an expectation is created by the metaphor, but the system behaves against that expectation. Metaphor can restrict a system that could otherwise explore new control interfaces. It can also confuse the player and the audience when the sound interface cannot be adequately created because of technical constraints.

There are many future directions for this research work. Direct extensions to the system could include more complex mappings involving additional parameters such as variable drop types or sizes, and waveform sculpting to allow the player to control granules’ attack and sustain. The concept of metaphoric instruments can be explored both within the class of instruments based on particle simulation for granular synthesis and in other classes.

In our last example of metaphor, we look at a system that uses a vocal tract metaphor in a gestural controller for speech synthesis.

6. GLOVE-TALKII: A METAPHOR FOR SPEECH SYNTHESIS

One of our most expressive instruments is voice. Here both player and audience are experienced speakers and listeners. Voice allows for some of the most expressive capacity of humans in both content and form. In Glove-TalkII (Fels and Hinton 1998), we developed a system

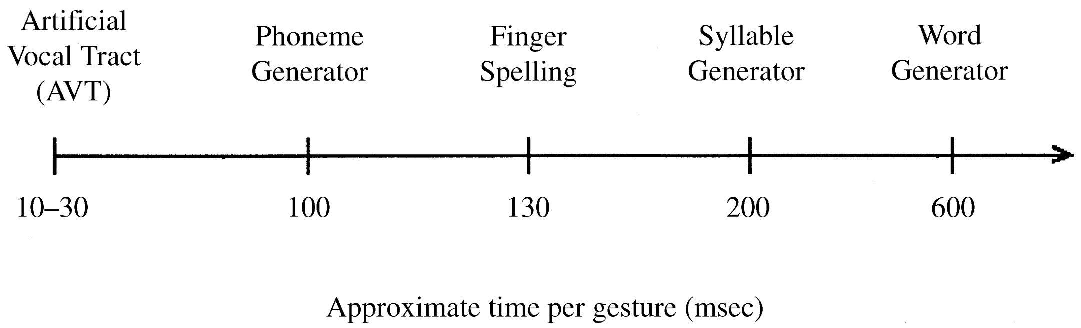


Figure 10. Spectrum of gesture-to-speech mappings based on the granularity of speech.

to allow a speaker to speak with a new instrument controlled with her hands and feet. We anticipated that the control bandwidth necessary for this task would be quite high so it was critical to make the system as transparent as possible for the speaker. Finally, as the actual speech synthesizer's control space was mostly formant frequencies and amplitudes, we required a system that could map between the easy-to-understand metaphor space and the formant space. Thus, we had three main concerns in developing the interface:

(1) Create a clear, easy to understand metaphor for speech production.
(2) Adapt mapping to match the speaker's interpretation of the metaphor as well as to maintain the integrity of the metaphor.
(3) Provide mechanisms to map from the speaker's cognitive space (which is based on metaphor) to the formant space of the speech synthesizer.

The first task required to build Glove-TalkII was creating an easy to understand metaphor. For this, we used an articulatory model of speech over other possible schemes. Many different possible mappings exist for converting hand gestures to speech. The choice of mapping depends on the granularity of the speech that you want to produce. Figure 10 identifies a spectrum defined by possible divisions of speech based on the duration of the sound for each granularity. What is interesting is that in general, the coarser the division of speech, the smaller the bandwidth necessary for the speaker. In contrast, where the granularity of speech is on the order of articulatory muscle movements (i.e. the artificial vocal tract (AVT)) high bandwidth control is necessary for good speech. The metaphor for this mapping suggests gesture is like vocal articulation. The AVT allows unlimited vocabulary, control of pitch and non-verbal sounds. Glove-TalkII is an adaptive interface that implements an AVT.

The second task, once we decided upon using an articulatory model of speech production as a metaphor, required developing a gestural mapping relating hand gesture to speech articulation. The representation we settled on is described in subsection 6.1. One of the important features of this space is that most of the mapping is continuous.[6] That is, there are no classification boundaries for the different types of vocal sounds. This allows the speaker to have all the expressive power of a normal voice. With this approach it is possible for a speaker to sing, speak different languages and make non-verbal sounds. The overall functionality of the system and the potential intimacy with the voice is increased.

The third task, once the gestural mapping was defined, was to actually build a computational system to implement the mapping. Note that the speaker is manipulating speech in the articulatory domain but the speech synthesizer works in the formant frequency domain. Thus, the Glove-TalkII system had to map from the speaker's interpretation of the metaphor, that is, which hand gestures they thought produce which speech, to the actual formant frequency space. While this could have been statically done (hard coded), we needed to maximise the control bandwidth between a speaker's gestures and the control of the formant frequencies. Further, each speaker had differing gesture abilities and interpretations of the metaphor. Thus, an adaptive system was used with the map between gestures and speech learned with neural networks.

The mapping between the speaker's actions and the sound is governed both by static and adaptive maps using neural networks. The speaker's hand gestures are dictated by their interpretation of the metaphor. Thus, from their perspective they are controlling an articulatory speech synthesizer. The neural networks' role is to learn the mapping between the speaker's interpretation of the metaphor and the formant frequencies. Because the system adapts to the speaker's understanding of the metaphor, she can have an incomplete sense of the original metaphor. Her interpretation though does need to be consistent for the system to learn it. If successful, the mapping will be more easily made transparent for the speaker (and possibly the audience), as articulation space is considerably more transparent than formant

[6] The stop consonants are an exception as button presses are used to produce them.

space. The techniques used are described in the following section.

6.1. System overview

The Glove-TalkII system converts hand gestures to speech, based on a gesture-to-formant model. The gesture vocabulary is based on a vocal-articulator model of the hand. By dividing the mapping tasks into independent subtasks, a substantial reduction in network size and training time is possible (see Fels and Hinton 1993).

Figure 11 illustrates the whole Glove-TalkII system. Important features include the three neural networks labelled vowel/consonant decision (V/C), vowel, and consonant. The V/C network is a 12–10–1 feed forward neural network with sigmoid activation functions (Rumelhart, McClelland *et al.* 1986). The V/C network is trained on data collected from the speaker to decide the probability that he wants to produce a vowel rather than a consonant sound. Likewise, the consonant network is trained to produce consonant sounds based on speaker-generated examples from an initial gesture vocabulary. The consonant network is a 12–15–9 feed forward network. It uses normalised radial basis function (RBF) (Broomhead and Lowe 1988, Fels 2001) activations for the hidden units and sigmoid activations for the output units. In contrast, the vowel network implements a fixed mapping between hand-positions and vowel phonemes defined by the speaker. The vowel network is a 2–11–8 feed forward network. It also uses normalised RBF hidden units and sigmoid output units (Fels and Hinton 1995). Eight contact switches on the speaker's left hand designate the stop consonants (B, D, G, J, P, T, K, CH), because the dynamics of such sounds proved too fast to be controlled by the speaker. The foot pedal provides a volume control by adjusting the speech amplitude and this mapping is fixed. The fundamental frequency, which is related to the pitch of the speech, is determined by a fixed mapping from the speaker's hand height. The output of the system drives ten control parameters of a parallel formant speech synthesizer every 10 ms. The ten control parameters are: nasal amplitude (ALF), first, second and third formant frequency and amplitude (F1, A1, F2, A2, F3, A3), high frequency amplitude (AHF), degree of voicing (V) and fundamental frequency (F0).

Once trained, Glove-TalkII can be used as follows: to initiate speech, the speaker forms the hand shape of the first sound he intends to produce. He depresses the foot pedal and the sound comes out of the synthesizer. Vowels and consonants of various qualities are produced in a continuous fashion through the appropriate co-ordination of hand and foot motions. Words are formed by making the correct motions; for example, to say 'hello' the speaker forms the 'h' sound, depresses the foot pedal and quickly moves his hand to produce the 'e' sound, then the 'l' sound and finally the 'o' sound. The speaker has complete control of the timing and quality of the individual sounds. The articulatory mapping between gestures and speech is decided *a priori*. The mapping is based on a simplistic articulatory phonetic description of speech (Ladefoged 1982). The X, Y coordinates (measured by the Polhemus) are mapped to something like tongue position and height[7] producing vowels when the speaker's hand is in an open configuration (see figure 12 for the correspondence and table 2 for a typical vowel configuration). Manner and place of articulation for non-stop consonants are determined by opposition of the thumb with the index and middle fingers. Table 2 shows the initial gesture mapping between static hand gestures and static articulatory positions corresponding to phonemes. The ring finger controls voicing. Only *static* articulatory configurations are used as training points for the neural networks, and the interpolation between them is a result of the learning but is not explicitly trained. For example, the vowel space interpolation allows the speaker to easily move within vowel space to produce diphthongs.

6.2. Mapping and expression

With 100 hours of practice, the one speaker who learned to speak with Glove-TalkII was able to speak with expression and be intelligible. We hypothesise that one of the most important design decisions that made this possible was the use of an easy-to-understand metaphor that constrained the speech task. With the strong metaphor and the adaptive mapping, the Glove-TalkII system facilitated making the mapping between gesture space and speech transparent for the speaker. For the listener, the speech was intelligible and thus expressive implying some transparency. Further, the fact that a speaker's hand gestures were mapped one-to-one to the speech output suggests that the mapping was also transparent to the listener to some extent.

6.3. Glove-TalkII: lessons learned

When considering how to create transparent mappings for future controllers, several key points may be learned from the Glove-TalkII system:

(1) Make the initial mapping easy to understand. In the case of Glove-TalkII, this was achieved by using an articulatory metaphor for speech production and developing a gestural system to control speech articulation based on this metaphor. This provided:

- an easy-to-understand mapping for speakers who have normal vocal tract-based speech;
- an easy-to-teach mapping: instruction on how to

[7] In reality, the X,Y coordinates map more closely to changes in the first two formants, F1 and F2 of vowels. From the speaker's perspective though, the link to tongue movement is useful.

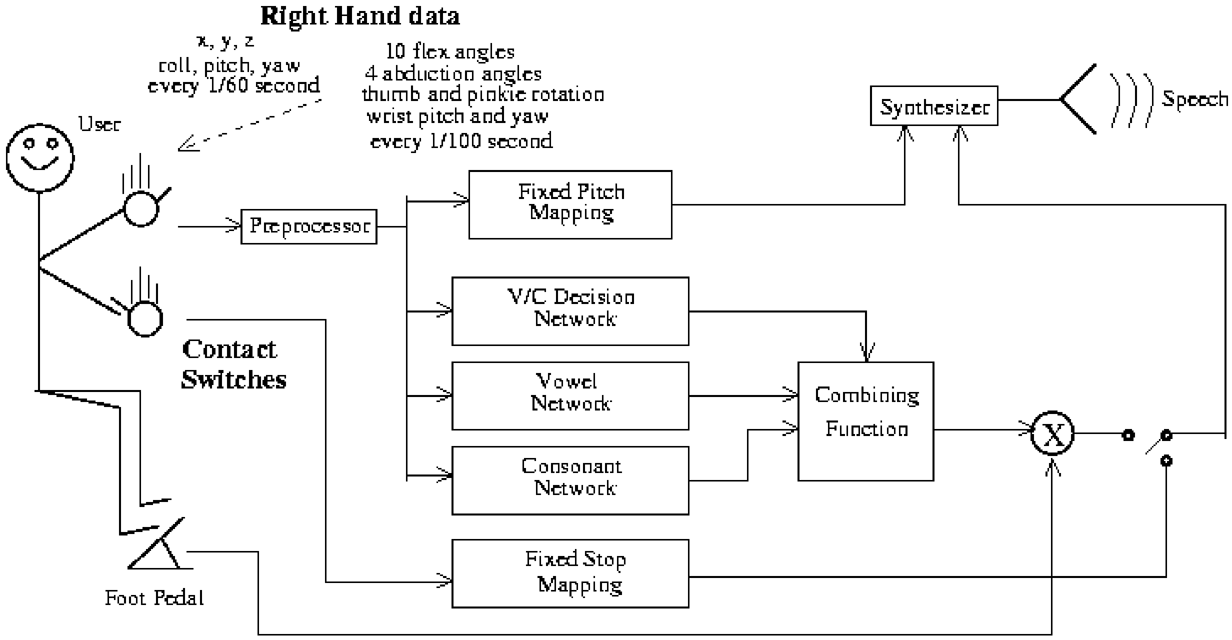


Figure 11. Block diagram of Glove-TalkII. Input from the speaker is measured by the CyberGlove, Polhemus, ContactGlove and foot pedal, then mapped using neural networks and fixed functions to formant parameters which drive the parallel formant synthesizer (Rye and Holmes 1982).

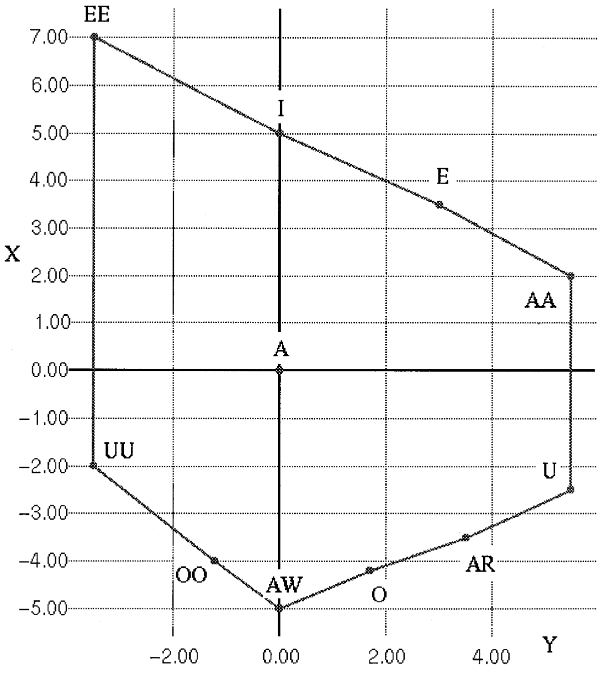


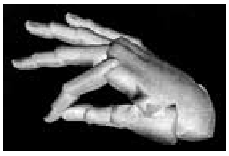
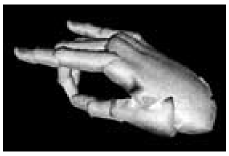
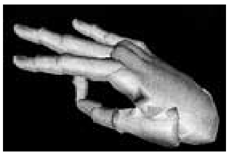
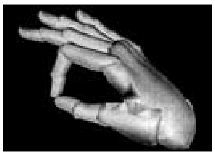
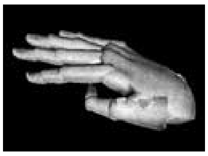
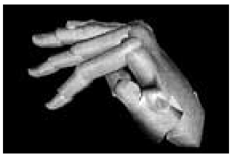
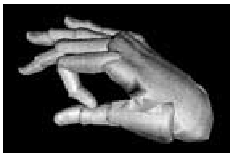
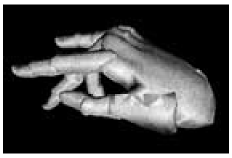
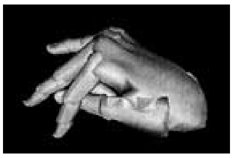
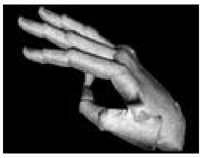
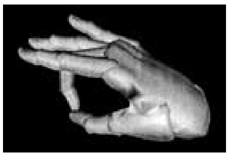
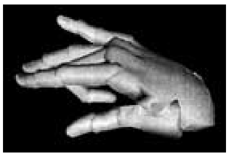
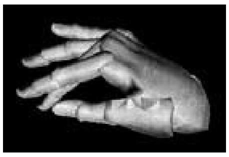
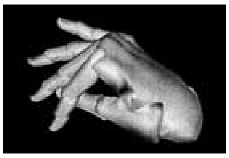
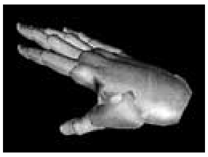
Figure 12. Hand-position-to-vowel-sound mapping. The coordinates are specified relative to the origin at the sound A.

start making sounds was simple and required little study;

- co-articulation of sounds maintained: by maintaining the metaphor which dictates a mostly one-to-one mapping between action and sound, the co-articulation effects in gesture space provided co-articulation in speech space, allowing for more diversity in the production of vocal sounds.

(2) Provide an adaptive mapping. Adding adaptive elements to the mapping allows the speaker to think and act in articulatory space even though the actual output space is not. This helps to increase transparency of the mapping for the speaker. Further, as the adaptive mapping learns the speaker's interpretation of the metaphor, the system maintains a consistent metaphor for the individual speaker. Adaptive

Table 2. Static gesture-to-consonant mapping for all phonemes. Note that each gesture corresponds to a static non-stop consonant phoneme generated by the text-to-speech synthesizer.

 **DH**	 **F**	 **H**	 **L**	 **M**
 **N**	 **R**	 **S**	 **SH**	 **TH**
 **V**	 **W**	 **Z**	 **ZH**	 **vowel**

elements have to be introduced carefully so that the mapping is not changing too quickly while the speaker is learning.

In summary, Glove-TalkII demonstrates that it is possible to design a system that translates hand gestures into speech using an artificial vocal tract model. With only 100 hours of training, a speaker's speech is intelligible and expressive. The use of an articulatory metaphor helped make the mapping transparent for the speaker. For the audience, part of the mapping is transparent in that they know what speech sounds like, thus enhancing the expressivity of the device. As the gestural system is based on speech production, it is possible that the entire mapping between gesture and speech could become part of the literature. It could provide a new gesture language that is expressive for both hearing and non-hearing communities as the relationship between sound and gesture will be transparent.

7. CONCLUSIONS AND FUTURE WORK

We introduced a two-axis transparency scale for understanding mappings and some of the conditions that make them expressive. Our need for this framework stems from the desire to design and build new instruments for musical expression. We want our framework to facilitate the acceptance of novel controllers into the literature to allow for new forms of expression. From this perspective, we propose that metaphor helps both the player and the audience make the mapping of a musical device transparent, hence making the device itself expressive. We also discussed how other methods facilitate making novel controllers and mappings more transparent.

We presented four examples of novel music and voice controllers that use metaphor as part of their mapping design strategies. In the parts of the mappings of these systems where the metaphor matches the implementation, we do see more expression. However, we also see the inherent difficulties with metaphor. In the Iamascope, the guitar metaphor helped with understanding the mapping, but could not compensate for the lack of tactile feedback that would be felt with a real guitar string. In Sound Sculpting, the use of a stretching metaphor overcomes the limitations of a lack of haptic feedback. However, places where arbitrary mappings are used break the metaphor, making some parts of the mapping opaque. Likewise, in MetaMuse, when granules behave like virtual rain the metaphor works. But once the discrete nature of the samples is noticed it becomes apparent to the musician that the metaphor is not working. Glove-TalkII circumvented this breakdown of the metaphor by adapting the mapping to be whatever the player thought the metaphor was. This works well only if the player's initial understanding of the metaphor is consistent and spans the whole range of outputs. An initial mapping based on a strong, easy-to-learn metaphor helped establish this criterion.

The main guideline when using metaphor for design is to use it as a stepping-stone for players and audiences. When the metaphor is not consistent the designer should provide enhanced functionality that is directly accessible. The enhanced functionality allows the performer to explore new sounds, providing the motivation to learn the unfamiliar controls. Using a convergent non-modal mapping provides transparency for the player and audience when encountering new technology. However, convergence should not be so great as to impede transparency for the audience – in some situations a one-to-one mapping may be more appropriate.

We have not addressed how to measure transparency or expressivity. Our belief is that transparency is correlated to cognitive load. This implies that the player or audience can handle an increasing number of (non-competing) cognitive tasks as the mapping becomes more transparent. Thus, we may be able to measure transparency using distractor tasks to load the player or audience. We are exploring this technique for determining *intimacy* with a device in human–human and human–computer interaction; it is left for future research. As for expressivity, by considering expression as a communicative act we can correlate players and audience responses to each other to measure expression. Experimental methods for this approach are very much in their infancy.

In summary, we believe that metaphor is an excellent stepping-stone for designing interfaces and mappings. The use of metaphor should facilitate bringing new, technology-driven interfaces and mappings into the literature as it increases transparency, thereby increasing expressivity. We caution, however, that metaphor is not a panacea and inherits all the good *and* bad qualities of the literature used as its basis.

ACKNOWLEDGEMENTS

The authors would like to thank the members of the Human Communication Technologies Lab for their continued support. We would also like to thank ATR and NSERC for their generous funding. Tom Calvert, Teena Carnegie, Kenji Mase and the members of the NIME 1 workshop all provided helpful discussion in the formation of these ideas.

REFERENCES

Bahn, C., and Trueman, D. 2001. Interface: electronic chamber ensemble. In *ACM SIGCHI Workshop on New Interfaces for Musical Expression*, electronic proceedings. Seattle, WA.

Broomhead, D., and Lowe, D. 1988. Multivariable functional interpolation and adaptive networks. *Complex Systems* **2**: 321–55.

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

Fels, S. S. 2001. Using normalized RBF networks to map hand gestures to speech. In R. J. Howlett and L. C. Jain (eds.) *Radial Basis Function Networks 2*, pp. 59–101. Physica-Verlag.

Fels, S. S., and Hinton, G. E. 1993. Glove-talk: A neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks* **4**: 2–8.

Fels, S. S., and Hinton, G. E. 1995. Glove-TalkII: mapping hand gestures to speech using neural networks. In G. Tesauro *et al.* (eds.) *Advances in Neural Information Processing Systems* **7**: 843–50. Denver, CO: MIT Press.

Fels, S. S., and Hinton, G. E. 1998. Glove-TalkII: A neural network interface which maps gestures to parallel formant speech synthesizer controls. *IEEE Transactions on Neural Networks* **9**: 205–12.

Fels, S. S., and Mase, K. 1999. Iamascope: a graphical musical instrument. *Computers and Graphics* **2**: 277–86.

Gadd, A., and Fels, S. S. 2002a. Metamuse: a novel control metaphor for granular synthesis. In *Proc. of the ACM Special Interest Group on Computer Human Interaction (SIGCHI '02),* pp. 636–7. Minneapolis, MN.

Gadd, A., and Fels, S. S. 2002b. MetaMuse: metaphors for expressive instruments. In *Proc. of New Interfaces for Musical Expression (NIME '02),* pp. 143–8. Dublin, Ireland.

Galyean, T. A. 1991. Sculpting: an interactive volumetric modeling technique. *ACM Computer Graphics (SIGGRAPH '91)* **25**(4): 267–74.

Hinckley, K., Pausch, R., Goble, J. C., and Kassell, N. F. 1994. Passive real-world interface props for neurosurgical visualization. In *Proc. of Computer-Human Interaction (CHI '94),* pp. 452–8. Boston, MA.

Hunt, A., Wanderley, M., and Kirk, R. 2000. Towards a model for instrumental mapping in expert musical interaction. In *Proc. of the Int. Computer Music Conf. (ICMC 2000),* pp. 209–12. Berlin, Germany.

Hunt, A., Wanderley, M., and Paradis, M. 2002. The importance of parameter mapping in electronic instrument design. In *Proc. of New Interfaces for Musical Expression (NIME '02),* pp. 149–54. Dublin, Ireland.

Institute de Recherche et Coordination Acoustique/Musique (IRCAM). 2002. http://www.ircam.fr/produits/logiciels/jmax-e.html

Ishii, H., and Ullmer, B. 1997. Tangible bits: towards seamless interfaces between people, bits and atoms. *Proc. of Conf. on Human Factors in Computing Systems (CHI '97)*, pp. 234–41. Atlanta, GA: ACM.

Jorda, S. 2001. New musical interfaces and new music-making paradigms. In *ACM SIGCHI Workshop on New Instruments for Musical Expression*, electronic proceedings. Seattle, WA.

Kramer, J., and Leifer, L. 1989. The 'Talking Glove': a speaking aid for nonvocal deaf and deaf-blind individuals. *Proc. of RESNA 12th Annual Conf.*, pp. 471–2.

Ladefoged, P. 1982. *A Course in Phonetics (2nd edn).* New York: Harcourt Brace Javanovich.

Lecuyer, A., Coquillart, S., Kheddar, A., Richard, P., and Coiffet, P. 2000. Can isometric input devices simulate force feedback? *IEEE Int. Conf. on Virtual Reality*, pp. 83–90.

Marx. A. N. 1994. Using metaphor effectively in user interface design. In *Proc. of Computer–Human Interaction (CHI '94),* pp. 379–80. Boston, MA.

McCaig, G., and Fels, S. S. 2002. Playing on heart-strings: experiences with the Two-Hearts system. In *Proc. of New*

Interfaces for Musical Expression (NIME '02), pp. 54–9. Dublin, Ireland.

Moore, F. R. 1988. The dysfunctions of midi. *Computer Music Journal* **12**(1): 19–28.

Mulder A., Fels, S. S., and Mase, K. 1999. Design of virtual 3d instruments for musical interaction. *Graphics Interface '99*, pp. 76–83.

Norman, D. A. 1998. *The Invisible Computer.* Cambridge, MA: MIT Press.

Norman, D. A. 1990. *The Design of Everyday Things.* Currency/Doubleday.

Ousterhout, J. K. 1994. *Tcl and the Tk Toolkit.* New York: Addison-Wesley.

Rokeby, D. 1995. Transforming mirrors: subjectivity and control in interactive media. In *Critical Issues in Electronic Media*, pp. 133–58. Albany, NY: SUNY Press.

Rumelhart, D. E., McClelland, J. L., and the PDP Research Group. 1986. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vols. 1 and 2.* Cambridge, MA: MIT Press.

Rye, J. M., and Holmes, J. N. 1982. A versatile software parallel-formant speech synthesizer. *Technical Report JSRU-RR-1016.* Joint Speech Research Unit, Malvern, UK.

Svanæs, D., and Verplank, W. 2000. In search of metaphors for tangible user interfaces. In *Proc. of DARE 2000: on Designing Augmented Reality Environments*, pp. 121–9. ACM.

Truax, B. 1988. Real-time granular synthesis with a digital signal processor. *Computer Music Journal* **12**(2): 14–26.

Trueman, D., and Cook, P. 1999. BoSSA: the deconstructed violin reconstructed. In *Proc. of the Int. Computer Music Conf. (ICMC '99)*, pp. 232–9. Beijing.

Winkler, T. 1995. Making motion musical: gestural mapping strategies for interactive computer music. *Proc. of the Int. Computer Music Conf. (ICMC '95)*, pp. 261–4. Banff, Canada.