Design of Virtual 3D Instruments for Musical Interaction

Axel G.E. Mulder
School of Kinesiology, Simon Fraser University, Burnaby, BC, Canada V5A 1S6. Email amulder@sfu.ca
S. Sidney Fels
Dept. of Electrical and Computer Eng., UBC, Vancouver, BC, Canada V6T 1Z4. Email ssfels@ece.ubc.ca
Kenji Mase
ATR MI & C Research Labs, Seika-cho, Soraku-gun, Kyoto, 619-02 Japan. Email mase@mic.atr.co.jp

Abstract

An environment for designing virtual instruments with 3D geometry has been prototyped and applied to realtime sound control and design. It was implemented by extending a realtime, visual programming language called Max/FTS, running on an SGI Onyx, with software objects to interface CyberGloves and Polhemus sensors and to compute human movement and virtual object features. Virtual input devices with behaviours of a rubber balloon and sheet were designed for the control of sound spatialization and timbre parameters. Informal evaluation showed that a sonification inspired by the physical world appears natural and effective. More research is required for a natural sonification of virtual input device features such as shape, taking into account possible co-articulation of these features. While both hands can be used for manipulation, left-hand-only interaction with a virtual instrument may be a useful replacement for and extension of the standard music synthesizer keyboard modulation wheel. More research is needed to identify and apply manipulation pragmatics and movement features, and to investigate how they are co-articulated, in the mapping of virtual object parameters.

Key words:human-computer interface, multidimensional control, virtual sculpting, sound editing, multimedia mapping, musical instrument design, gesture interface.

1 Introduction

We report in this paper on work in progress to develop adaptable gestural interfaces for simultaneous multidimensional control [5]. An example of simultaneous multidimensional control can be found in music composition and sound design, which task involves the manipulation of many inter-dependent parameters simultaneously. For any sound designer or composer and certainly a performer the control of these parameters involves a significant amount of motor and cognitive processing to coordinate his or her motor system when mouse-and-keyboard interfaces are used. These interfaces capture only a very limited range of gestural expressions and are not very adaptable to the gestural preferences or motor capabilities of a user. Although the human hand is well-suited for multidimensional control due to its detailed articulation, the mouse and keyboard do not fully exploit this capability. In fact, most gestural interfaces, even those that use a dataglove-like interface do not fully exploit this capability due to a lack of understanding of the way humans produce their gestures as well as a lack of understanding what meaning can be inferred from these gestures [7].

Thus, to reduce the motor and cognitive load for the sound designer, it is necessary to design a gestural interface that implements data reduction with respect to the controlled parameters and/or an interface that can be adapted to exploit the capability of human gestures to effortlessly vary many degrees of freedom simultaneously at various levels of abstraction. In terms of data reduction of sound synthesis parameters, we use a sound synthesis environment that facilitates representation at various levels of abstraction of sound. In terms of exploiting human gestural capability, we use a virtual input device that can be programmed to sense and respond to manipulation by the entire hand and as such allow for a wide range of gestural expressions to have effect.

In addition, by creating intuitively related representations of the feedback from the gestural expression in various media, we hope the user's perceptual system will have more information about the control space and thus be able to decide faster between the different control possibilities. The general aim of this research is to create a multiple abstraction level, consistent and unified, yet user-adaptable input method for simultaneous multidimensional continuous control tasks such as sound design.

1.1 Sound Sculpting

We are evaluating the above approach by implementing sound sculpting [6] as a design environment for virtual musical instruments. In sound sculpting, a virtual object is used as input device for the editing of sound - the sound artist literally "sculpts" sounds using a virtual sculpting computer interface [4], i.e. by changing virtual object parameters such as shape, position and orientation. In our study, the object is virtual, i.e. the object can only be perceived through its graphics display and acoustic representations, and has no tactile representation. Although sculpting in the physical world is most effective with touch and force feedback, our assumption is that these forms of feedback can be replaced by acoustic and visual feedback with some compromises. The motivation to make such assumptions is based on the fact that the generation of appropriate touch and force feedback, while exploiting the maximum gestural capability in terms of range of motion and dexterity, is currently technically too challenging in a virtual object manipulation task.

1.2 Gestures

We are focusing on physical manipulation gestures, such as used for changing object shape (i.e. sculpting), in which the shape, position and orientation of the hand is changing simultaneously. Such gestures provide the highest dimensionality of control, especially when two hands are involved in the manipulation. In addition, if well designed, two-handed manipulation increases both the directness and degree of manipulation of the interface [1], thereby reducing the motor and cognitive load. We are also interested in the use of dynamic signs (hand shape is constant, but hand position and/or orientation is changing), although they represent less dimensions of simultaneous control. The other end of the spectrum, in terms of control dimensionality, is represented by static signs, which are not useful in the task we are interested in (i.e. multidimensional control) other than for selection tasks. In previous work on gesture interfaces such as [3] it has been noted that, since humans do not reproduce their gestures very precisely, natural gesture recognition is rarely sufficiently accurate due to classification errors and segmentation ambiguity. Only when gestures are produced according to well-defined formalisms, such as in sign language, will automatic recognition have acceptable precision and accuracy. However, a gesture formalism will require tedious learning by the user. Thus we do not compute or analyse any abstract symbolic representation of the gestures produced by the user, but instead focus on the continuous changes represented by the gestures.

1.3 Pragmatics

When we consider object manipulation as the changing of position, orientation and shape of an object, the pragmatics for position and orientation changes of small, light objects are simple and do not involve any tools. However, an analysis of the methods employed by humans to edit shape with their hands leads to the identification of three different stereotypical methods.

- **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.
- **Carving** - The shape of objects made of material with medium stiffness, like many wood materials, are often changed by holding the object in one hand and applying forces to the object using a tool like a knife or a file.
- **Chiseling** - The shape of objects made of material with high stiffness, like many stone materials, are often changed by placing the object on a supporting surface and applying forces to the object using a tool like a chisel held in one hand and a hammer held in the other.
- **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(s) place a pre-shape component.

2 Mapping Strategies

Sound sculpting should minimize the motor and cognitive load, or in other words be “easy to use”, for a novice user of the system. The ease-of-use is determined by the design of the mapping. Figures 1 and 2 through 4 illustrate our approach of using virtual object parameters at various levels of abstraction as a means to relate hand movements to sound variations.

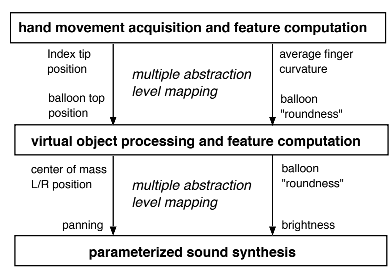

Figure 1: Functional diagram of hand movement to sound mapping. Virtual object features are used as a means to relate hand movement features to sound features.

2.1 Abstraction Levels

Our aim is to address the fact that hand movements, virtual object parameters such as shape, and sound are all multidimensionally parameterized at various levels of abstraction. We hope that explicit access by the user to these levels of abstraction will facilitate design and use of a mapping of human movement to sound. We also hope that a mapping will be faster to learn when movement features are mapped to sound features of the same abstraction level. However, as stated in the introduction, the mouse and keyboard capture only a specific range of hand movements, so that computation of gestural expression at levels of abstraction other than the physical level is always based on that specific range of movements. Hence, the number of gestural expressions represented at other levels of abstractions is only a small part of the full range of gestural expressions. On the contrary, by presenting the input device to the user as a virtual object that can be programmed to respond to almost any hand movements, representations at other levels of abstraction that are of interest become available.

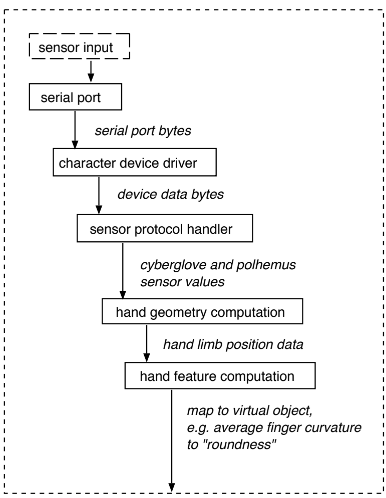

Figure 2: Functional diagram of hand movement acquisition and feature computation. The diagram shows mapping at the feature abstraction level, but mapping at other levels is equally possible.

2.2 Mapping Examples

In terms of relating the parameters of a virtual object to sound, a mapping based on the real, physical world may be easy to understand, but will not provide suggestions for the control of abstract, higher level sound parameters. Nevertheless, if the object were taken as a sound radiating object, a simplified mapping of the virtual object's position and orientation to sound parameters could involve mapping left/right position to panning, front/back or depth position to reverb level, angle of the object's main axis with respect to the vertical axis to reverb time

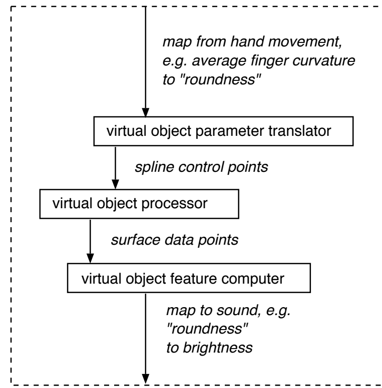

Figure 3: Functional diagram of virtual object processing and feature computation. The diagram shows mapping at the feature abstraction level, but mapping at other levels is equally possible.

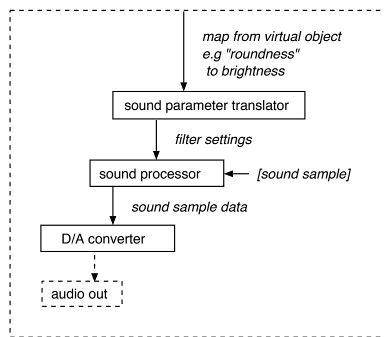

Figure 4: Functional diagram of sound synthesis. The diagram shows mapping at the feature abstraction level, but mapping at other levels is equally possible.

and virtual object volume to loudness. Mapping virtual object shape parameters to timbral parameters is less easily based on the physical world, but could involve mapping the length of the main, i.e. longest, axis to flange amplitude, transverse surface area or object width to chorus depth and curvature or "roundness" to brightness. While we are currently investigating mapping to sound we are interested in applying the developed interaction methodology to other domains, such as the editing of texture, color and lighting of graphical objects.

2.3 Explicit vs. Implicit

The above approach involves explicit knowledge of the mapped parameters, however, another approach would involve implicit knowledge. Such an approach can be implemented with neural networks, which require that the user "teach" the system which hand movements it should map to the sound parameters of interest [2]. In both ways the system can be adapted to the user's preferences, but the implicit method does not provide any guidelines when designing a mapping, which may be an advantage in situations where the user already has an idea of how a sound should change when making a movement but a disadvantage in other situations where guidelines for the design of a mapping are desired or when presets needed.

3 Implementation

In the development of our prototyping environment we aimed to facilitate quick prototyping and experimentation with a multitude of gestural analysis methods by creating a set of tools that facilitate the computation of various gesture features and parameters.

3.1 Hardware Environment

Figure 5 illustrates the hardware device setup. We use an SGI Onyx with four R10000 processors and audio/serial option (6 serial ports) to interface two Virtual Technologies Cybergloves (instrumented gloves that measure hand shape - see also [8]) and a Polhemus Fastrak (a sensor for measuring position and orientation of a physical object, such as the human hand, relative to a fixed point). A footswitch enabled "holding" of the virtual object. While the Cyberglove is probably one of the most accurate means to register human hand movements, we have found accurate measurement of thumb movement difficult due to the fact that the sensors intended for the thumb do not map to single joints but to many joints at the same time. Nevertheless, with a sufficiently sophisticated hand model it is possible to reach acceptable accuracy for our purposes, but as we had not programmed a calibration procedure based on a set of hand postures, the calibration is tedious as well as individual specific. For graphic display of the hands and the virtual object we use OpenInventor software.

3.2 Software Environment

We use a visual, interactive programming environment for real-time sound synthesis and algorithmic music composition called Max/FTS [9] (figure 6). We chose Max/FTS as our platform due to its real-time computation and visual programming capabilities, the access to various synthesis models implemented as editable patches and the fact that it runs on an SGI, thus needing no special sound cards. Max/FTS functionality is extended by linking in dynamic shared objects (DSO) at run-time. In order to facilitate quick and easy prototyping of various gestural analysis computations and allowing for application of the computations to different bodyparts we have developed new Max/FTS objects. We exploit the

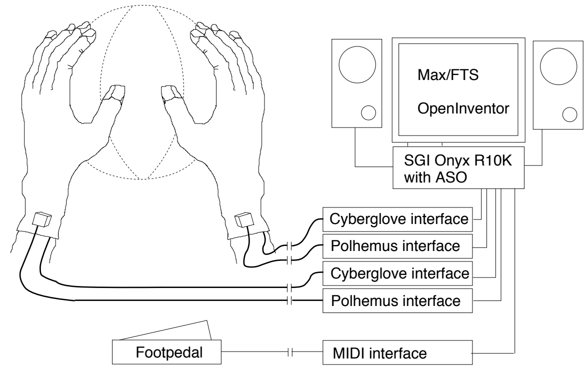

Figure 5: Hardware devices used. The dotted lines represent a virtual surface which the sound designer or sound composer manipulates.

strong datatype checking of Max/FTS to introduce new datatypes such as *position* and *orientation* for geometric and kinematic computations as well as a *voxel* and *hand* datatype. This way the user cannot apply objects at an inappropriate abstraction level, which would be possible if computed values were only represented as number data types. Real-time performance was achieved by using memory that was shared between two Max/FTS applications, one for sound synthesis and the other for remaining computations, to exchange data. The available sound parameters in our simplified synthesis model were loudness, spatialization parameters panning, reverberation time and level and timbre specific parameters attack time, release time, flange index, chorus depth, frequency modulation index, low/mid/high-pass filter amplitude and frequency.

3.3 New Max/FTS objects

The new Max/FTS software object we have programmed are listed below with reference to figures 1 and 2 through 4.

- **Unix character device drivers** - Objects for interfacing peripheral devices that communicate using one of the SGI's serial ports ($serial$), and for communication between Max and other processes using TCP sockets ($client$) or shared memory with semaphores (sms and smr). We are using the sms object amongst others to communicate data to an OpenInventor 3D graphics server to display the acquired hand shape, hand position and hand orientation data as a graphic hand as well as to display the virtual object graphically.

- **Sensor interfaces** - These objects are peripheral (character) device interfaces that implement a spe-

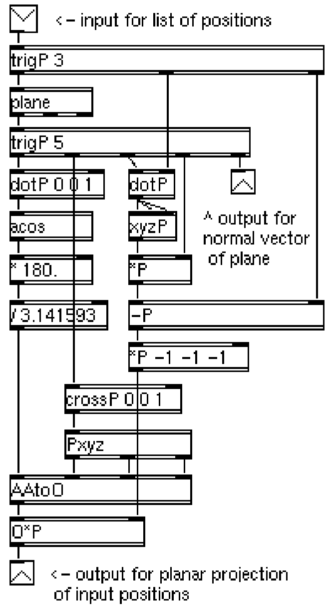

Figure 6: Typical example of geometric computing using the new Max/FTS *position* and *orientation* objects. This Max patcher takes a list of *positions*, projects these on the plane that best fits them, then rotates the plane (with projected *positions*) and eliminates z. 2D parametric curves can then be fitted to e.g. the x-y coordinates computed from the *positions* marking the thumb and index fingers).

cific protocol, such as for the Cyberglove and Polhemus sensors ($cyberglove$ and $polhemus$).

- **Geometric computation** - New datatypes *position* (x, y and z position as floats in one data structure) and *orientation* (a data structure containing a rotation matrix, euler angles and angle-axis representations of orientation) were defined for this group of objects to facilitate computations such as distance between points ($+P$, $-P$), scaling ($*P$), dot product ($dotP$), cross product ($crossP$), norm ($normP$), magnitude ($||P$), rotation ($O*P$ and $*O$), frame of reference switching (TO) etc. and their derivatives (see figure 6).

- **Geometric structure computation** - At this level the computations do not involve just points with a position and an orientation but involve volume elements (represented as a new datatype *voxel*). Voxels may be ordered and linked in a specific manner so as to represent for instance human anatomical structures or otherwise shaped objects. A new datatype *hand* is used to represent a human hand. The object $geoHand$ computes an ordered list of *voxels*, packed as a *hand* that are relative to the Polhemus transmitter from Cyberglove joint data and Polhemus receiver position and orientation data. This computation could also be done using the geometric computation objects in a patcher, but it will be computationally more expensive.

- **Hand feature computation** - Objects for the computation of hand features such as the distance between thumb and index fingertips and the distance between left and right hand palm are easily calculated using the above geometric computation objects in a patcher. We also made objects for computation of the orientation of a plane ($plane$, see also figure 6) such as the palm, the average position of a selected number of points of the hand ($avgP$) and for computation of features based on the path of a selected point of the hand ($bufP$). We intend to make other objects for the computation of features such as finger and hand curvature using a curve fitting algorithm, estimated grip force using a grasping taxonomy etc..

- **Virtual object processing** - We have programmed $sheet$, a physical model of a sheet of two layers of masses connected through springs (see figure 7). The four corners of the sheet function as the control points and can be stuck, or clamped to e.g. the index and thumb tips of both hands. Similarly, we have programmed $balloon$, a physical model of a single layer of masses positioned in sphere-like form (see figure 8).

- **Virtual object feature computation** - Many of the geometric computation objects and some of the hand feature objects can be used to compute virtual object features. Other types of virtual object feature computations involve superquadrics, a mathematical method to describe a wide variety of shapes with two parameters specifically related to virtual object (vertical and horizontal "roundness" or "squareness") and 9 others for size, orientation and position of the virtual object. Based on [10] we have programmed a $superquadric$ object that fits a virtual object described in terms of superquadric parameters to a set of *positions*. As the fitting process is iterative, it is computationally expensive and as yet too slow (order of 100 ms) for real-time control of timbre. A simpler approach is implemented in a feature computation object $ellipsoid$ which fits only a virtual object described in terms of ellipsoid parameters (i.e. three size parameters) to the list of *positions* and computes within real-time. For 2D curve fitting we have programmed a $polynomial$ object, which will approximate a list of *positions* by a polynomial of at most second degree. The $kappa2D$ and $kappa3D$ objects compute the curvature of a 2D curve respectively 3D surface.

4 Virtual Object Manipulation

We have created two types of virtual objects, and experimented with manipulation methods based on the pragmatics of claying. The pragmatics of claying consisted of manipulation of the virtual object permanently stuck to (a selected number of) joints and manipulation by "touching" the virtual object. A footswitch enabled selection of either manipulation method.

4.1 Manipulating a Rubber Sheet

We have experimented with the manipulation of a virtual object with a shape and behaviour of a rubber sheet. The object $sheet$ was used to compute the virtual object *voxels*. The *voxels* of the tips of the index and thumb fingers were used as the *voxels* of the four corners of the virtual sheet. Thus, rotating, but not moving, the finger tips would result in bending of the virtual sheet. The computations could be completed near to real-time if a suitably low number of virtual masses was chosen. Fingertips of both hands or one hand could be clamped to the corners of the sheet (figure 7). If only one or none of the hands was unclamped, the other hand or both hands could "touch" the sheet.

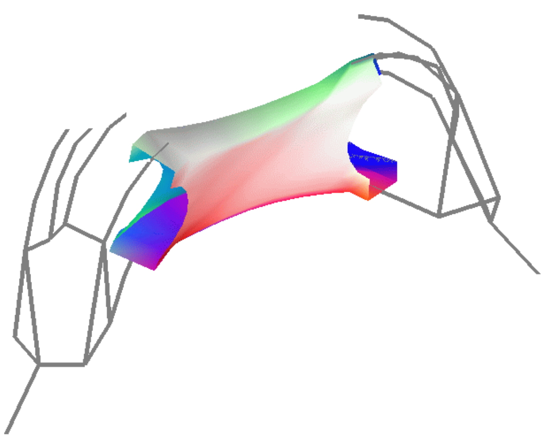

Figure 7: Example of the sheet clamped to the index and thumb tips of both hands.

4.2 Manipulating a Balloon

We have also experimented with the manipulation of a virtual object with a shape and behaviour of a rubber balloon. Manipulation of this virtual object occurred through the movement of any of the *voxels* that make up a hand. When the virtual object was clamped to the hands, a superquadric shape was fitted to the *positions* of these *voxels*. When switching from clamping to "touching", the virtual object surface *positions* were used as the starting positions for a physical simulation of a rubber balloon. Joints of both hands or one hand could be clamped to surface points of the balloon (figure 8). If no hand or only one hand was unclamped, the other hand(s) could "touch" the sheet.

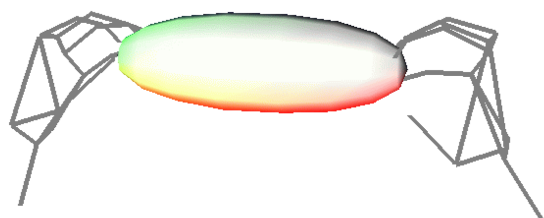

Figure 8: Example of the balloon clamped to both hands.

5 Evaluation

In our pilot studies to date, pitch and duration of the sound were either fixed or pre-programmed in a MIDI sequence. Mapping virtual object height to pitch could be "intuitive". However, due to measurement latency in the acquisition of the height of the virtual object, controlling pitch in this way was ineffective. Instead the height was mapped to mid-pass filter amplitude. We found that when mapping a virtual object feature to a sound parameter, offsetting and scaling the value of the virtual object feature required some arbitrary heuristics based on workspace dimensions etc.. A solution without such heuristics would be preferred. In the following two subsections we discuss the manipulation and mapping of the sheet and balloon virtual objects.

5.1 The Rubber Sheet

We evaluated the rubber sheet using the mapping of virtual object position and orientation as described in the section on mapping examples and the following mapping of virtual object shape. Average length of the sheet, along the axis between left and right hand was mapped to flange index. Width (i.e. axis between index and thumb) of the sheet was mapped to chorus depth. A measure of the average curvature, computed with $kappa3D$, was mapped to the frequency modulation index. The angle between left and right edge of the sheet was mapped to vibrato. The mapping of virtual object position and orientation took only a few moments to get used to. As for shape, it was found that curvature was difficult to control given the fact that only the four corners of the sheet could be manipulated. Normally, one positions fingers in various ways on the surface to control the curvature. The length was easiest to control, then the width and the angle between left and right edge of the sheet. Manipulation of the sheet itself with the indexes of both hands clamped to the sheet was very natural - there was very little effort required to master control. We also tried manipulation with the left hand only, similar to the familiar keyboard modulation wheel but with increased functionality. This form of manipulation with the left corners of the sheet clamped to the left hand index and thumb and the right corners of the sheet fixed in space, allowed gestures to be more expressive than typical keyboard modulation wheel gestures and hence provided for a dramatic performance effect. In addition, the reference point (the right corners) could easily be moved, so as to accomodate rapid changes in the zero-point of any modulations given different musical context, by clamping the right corners to the right hand temporarily. Manipulation using "touching" only was difficult, mainly due to the lack of tactile feedback.

5.2 The Rubber Balloon

We evaluated the rubber balloon using the mapping of virtual object position and orientation as described in section 2.2. As in the case of the sheet, this mapping worked equally well. We evaluated the following mapping of virtual object shape. The length of the main, i.e. longest, axis of the balloon was mapped to flange index and the cross sectional surface area was mapped to chorus depth. The objects $ellipsoid$ and $superquadric$ were used to compute the virtual object features. Informal evaluation showed this mapping to work well. "Roundness" of the object, estimated using the bending of the fingers was mapped to the frequency modulation index. Length was easiest to control and then cross sectional surface area and "roundness". This aspect of the mapping was more difficult to use due to occasional "manipulation cross-over", i.e. when the cross-sectional surface area was increased the balloon would also become more rectangular instead of ellipsoid, without the sound sculptor intending it. Another problem with this mapping is that it is unclear how the sound sculptor conveys "roundness" of the virtual object when changing the shape of his or her hands, due to the fact that normally not the entire hand is touching an object's surface. Any sphere-like object can be held with anywhere from a few finger tips to the entire hand. In addition, due to the fitting process, jitter occurred, i.e. the virtual object would at times jump from cube-like to ellipsoid without any significant hand motion, resulting in unpredictable sound variations. Possibly this can be circumvented by algorithm and Max/FTS patch adaptations. Manipulation using the left hand only, with the balloon entirely clamped to the hand was not as natural as in the case of the sheet, most likely due to a missing reference point (the right corners in the case of the sheet) in our case (the balloon could also be fixed in space at one end, but we haven't implemented this) and hence more difficulty in manipulating the shape of the balloon. Nevertheless manipulation of the position and orientation of the virtual object was effective, but the same effect may have been achieved without the presence of the virtual object. For similar reasons as in the case of the sheet, manipulation using "touching" only was difficult.

5.3 Summary of Evaluation

Based on informal testing by the author, approximately 15 research colleagues and video material of the author's use of the environment, the evaluation of the VMI environment can be summarized as follows:

- **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 performers advantage in certain tasks, 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.

- **Sonification** - The mapping of position and orientation to spatialization 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 user. 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 they were mapped to. Scaling and offsetting of virtual object features for mapping to sound parameters was somewhat arbitrary.

- **Adaptation** - Although adaptation of VMIs was possible in many diverse ways, the user interface to implement these adaptations was not so easy to use without significant technical expertise.

- **Engineering** - While acceptable real-time performance capturing almost all hand gestures, was achieved, expensive technology is required to implement it.
- **Interaction** - Different types of interaction were tried out. The left-hand-only manipulation method could provide a useful alternative to the standard keyboard modulation wheel in situations where the right hand is needed for pitch control using e.g. a keyboard. The naturalness of the interaction is affected by both the manipulation methods afforded by the virtual object and the sonification of the object and appears natural if inspired by the physical world.

6 Conclusion

The main result of the research is that a prototype of an environment for design of virtual instruments with 3D geometry has been successfully implemented, allowing real-time performance in 3D and adaptation to exploit many manipulation gestures.

While this result is encouraging, the preliminary evaluation of the environment showed that for a natural interaction to take place the user or performer should be able to:

- 1. apply manipulation methods used in everyday life to change shape, position and orientation of the virtual object
- 2. believe that the sonification of the virtual object is qualitatively and directly related to the variations in virtual object shape position and orientation.

In terms of human factors and gestural communication research, the work has shown the feasibility and usefulness of 3D interactive virtual input devices in a sound control task - further work is needed to demonstrate usefulness in other application domains. The value of the VMI environment as a virtual input device prototyping tool should not be underestimated.

Future work should address an important question that arose from this research: To what extent do performers really want their musical instruments adapted during their career, if they were given unlimited freedom to require such adaptation ? This question can be answered by experimentation with a musical instrument design environment as was implemented for the research presented in this paper. While the current environment focusses on hand movements, future work may also address other types of human movement.

Furthermore, future research on VMI's should concentrate on the identification of manipulation pragmatics, and their derived forms which are present in gesticulation and sign languages, as well as how movement features are co-articulated. Equal effort should be spent on identifying methods to visualize sound in terms of features of a 3D virtual object. This understanding will contribute to the creation of useful, effective and enjoyable new interaction methods. Engineering efforts should concentrate on the implementation of virtual object simulation techniques and faster sensor data acquisition, while whole-hand tactile and force feedback are most desirable for realization of more refined and immediate interaction.

7 References

[1] Paul Kabbash, William Buxton and Abigail Sellen, "Two-Handed Input in a Compound Task," *Proceedings of CHI '94*, pp 417-423, 1994.

[2] S. Sidney Fels and Geoffrey E. Hinton, "Glove-TalkII: Glove-TalkII: A neural network interface which maps gestures to parallel formant speech synthesizer controls," *IEEE Transactions on Neural Networks*, Vol. 9, No. 1, pp. 205-212, 1998.

[3] S. Sidney Fels and Geoffrey E. Hinton, "Glove-Talk: a neural network interface between a data-glove and a speech synthesizer," *IEEE Transactions on Neural Networks*, Vol. 4, No. 1, pp. 2-8, 1993.

[4] Tinsley A. Galyean, "Sculpting: An Interactive Volumetric Modeling Technique," *ACM Computer Graphics*, Vol. 25, No. 4, (SIGGRAPH '91, Las Vegas, 28 July - 2 August 1991), pp. 267-274, 1991.

[5] Axel G.E. Mulder. *Design of virtual 3D instruments for sound control*. PhD. thesis. Burnaby, BC, Canada: Simon Fraser University, 1998. Available on the web at http://www.cs.sfu.ca/~amulder/personal/vmi/AM98-thesis.ps.Z and http://www.cs.sfu.ca/~amulder/personal/vmi/AM98-thesis.pdf.

[6] Axel G. E. Mulder, "Getting a GRIP on alternate controllers: Addressing the variability of gestural expression in musical instrument design," *Leonardo Music Journal*, Vol. 6, pp. 33-40, 1996.

[7] Axel G. E. Mulder, "Hand gestures for HCI," *Technical Report, NSERC Hand Centered Studies of Human Movement project*, Burnaby, BC, Canada: Simon Fraser University, 1996. Available through the WWW at http://www.cs.sfu.ca/~amulder/personal/vmi/HCI-gestures.htm

[8] Axel G. E. Mulder, "Human Movement Tracking Technology," *Technical Report, NSERC Hand Centered Studies of Human Movement project*, Burnaby, BC, Canada: Simon Fraser University, 1994. Available through the WWW at http://www.cs.sfu.ca/~amulder/personal/vmi/HMTT.pub.html

[9] Miller Puckette, "FTS: A real time monitor for multiprocessor music synthesis," *Computer music journal*, Vol. 15, No. 3, pp. 58-67, 1991.

[10] Franc Solina and Ruzena Bajcsy, "Recovery of parametric models from range images: the case for superquadrics with global deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, No. 2, February, pp. 131-147, 1990.