音楽情報科学 23-12
(1997. 12. 16)

Mapping virtual object manipulation to sound variation

Axel G. E. Mulder, S. Sidney Fels and Kenji Mase

ATR Media Integration and Communications laboratories
Seika-cho, Soraku-gun, Kyoto, 619-02 Japan

Email mulder@mic.atr.co.jp, Tel. +81 774 95 1496

Abstract: We are studying the use of dexterous manipulation in a virtual environment to create, edit and perform sounds. In our Max/FTS-based experimental environment, a virtual object functions as input device for the editing of sound - the sound artist literally "sculpts" sounds by changing virtual object attributes (shape, position and orientation). We identified three sculpting methods and discuss one for use in physical and abstract mapping strategies. In pilot mapping experiments, the main problems we encountered were the search for interaction metaphors that can quickly be understood, incorporation of manipulation pragmatics in mappings, "touching" the virtual object and real-time computability.

1 Introduction

We report in this paper on work in progress to develop gestural interfaces that allow for simultaneous multidimensional control [4]. 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 sound composer the control of these parameters involves a significant amount of cognitive processing to coordinate his or her motor system when mouse-and-keyboard interfaces are used. The mouse and keyboard capture only a very limited range of gestural expressions. Although the human hand is well-suited for multidimensional control due to its detailed articulation, the mouse and keyboard do not exploit this capability. In fact, most gestural interfaces, even those that use dataglove-like interface do not fully exploit this capability due to a lack of understanding of the way humans produce their gestures and what meaning can be inferred from these gestures [6].

Thus, to reduce the cognitive load for the sound designer, it is necessary to design a human computer interface that implements data reduction with respect to the controlled parameters and/or an interface that exploits 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, sound synthesis models that parameterize timbre independently at a perceptual level are rather limited in their applicability and/or are computationally expensive. In terms of human gestural capability, we aim 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.

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.

2 Sound Sculpting

Our experimental aim is to implement sound sculpting as first outlined in [5]. In sound sculpting, a virtual object is used as input device for the editing of sound - the sound artist literally "sculpts" sounds by changing virtual object attributes 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.

2.1 Gestures

When hand shape, hand position and hand orientation are changing simultaneously such as in the variation of object shape through physical manipulation, the dimensionality of the control is highest. Although the variation of shape (i.e. 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 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.

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.

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.

2.2 Mapping Strategies

Figure 1 illustrates the approach we are taking in mapping human movement parameters to sound parameters. As stated in the introduction, the mouse and keyboard capture only a limited range of gestural expressions, so that computation of gestural expression at levels of abstraction other than the physical level is also limited. On the contrary, by presenting the input device as a virtual object to the user, gestural expression can more easily be captured and representations computed at various levels of abstraction. The motivation for distinguishing between abstraction levels is that they help the user when designing and using a mapping of human movement to sound. Both sound and human movement can be represented at various abstraction levels. We think a mapping will be faster

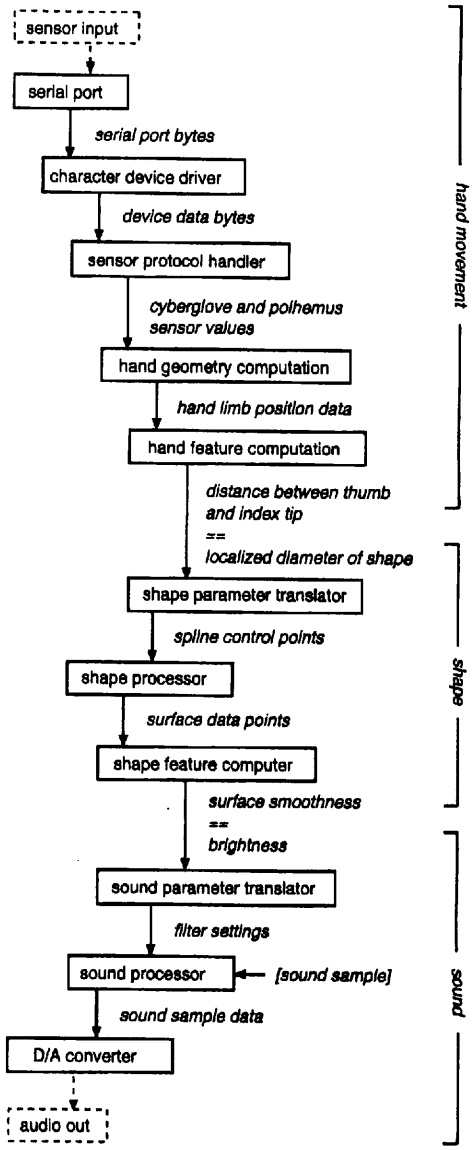

Figure 1: Functional diagram of hand movement to sound mapping. Virtual object attributes are used as a means to relate hand movement features to sound features. The diagram intends to illustrate mapping at a similar abstraction level of hand movement parameters to virtual object operators and of virtual object features to sound parameters. The example shows mapping at the feature abstraction level, but mapping at other levels is equally possible. While the hand feature and shape feature concepts appear in this diagram at the highest level of abstraction we are planning to develop other computations that use the features to derive higher abstractions of the hand and the virtual object.

to learn when movement features are mapped to sound features of the same abstraction level.

Thus, it is our strategy to use object attributes at various levels of abstraction as a means to relate hand movements to sound variations. Hand movements, object attributes such as shape, and sound are all multidimensionally parameterized at various levels of abstraction. We exploit the intuitive relations between shape (of physical objects) and timbre as well as between shape and manipulation for the design of a sound editing environment where the user can change the sound by applying operations such as the changing of shape to a virtual object.

Shape features are subsequently computed and mapped to sound parameters. Hand features that are computed from the acquired hand shape, hand position and hand orientation data using the above set of objects are mapped to shape parameters of a similar abstraction level. If a shape parameter is specified at a different abstraction level than the parameters of the shape processor, it will need to be translated to the parameter space of the shape processor. A shape processor is an algorithm that computes surface data points from shape parameter values. The simplest shape processor is the identity, i.e. it takes surface data points as shape parameters and outputs the same surface data points. If the hand feature computation and shape parameter translation are also taken as the identity, the 3-D positions that represent the hand will also specify the virtual object surface or in other words, the hands are always "touching" the virtual object. A shape feature is computed from surface data points and represents an abstraction of the shape.

While the above approach involves explicit knowledge of the mapped parameters, 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]. Either way, the system can be adapted to the user's preferences. We are furthermore investigating how to apply the developed interaction methodology to other domains, such as the editing of texture, color and lighting of graphical objects.

3 Implementation

3.1 The Environment

We are using Max/FTS [8] on an R10000 SGI Onyx with audio/serial option (6 serial ports) to interface two Virtual Technologies Cybergloves (instrumented gloves that measure hand shape - see also [7]) 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). Figure 2 illustrates the hardware device setup. While the Cyberglove is probably one of

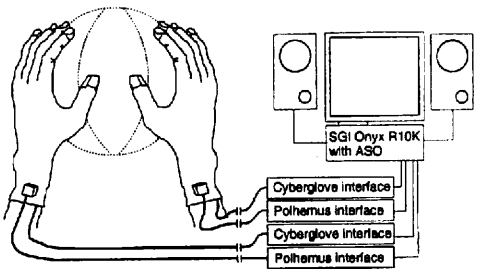

Figure 2: Hardware devices used. The dotted lines represent a virtual surface which the sound designer or sound composer manipulates.

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 the calibration is tedious as well as individual specific.

Max/FTS is a visual, interactive programming environment for real-time sound synthesis and algorithmic music composition. Figure 3 shows a typical Max project. 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 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.

3.2 New Max/FTS objects

The new Max/FTS software object we have programmed are listed below with reference to figure 1.

- **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*). We are using the *client* object amongst others to communicate data to

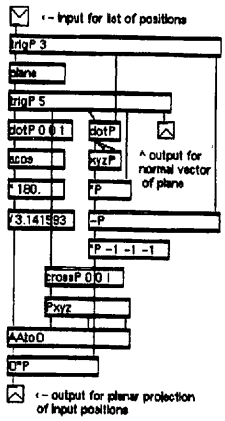

Figure 3: Typical example of geometric computing using the new Max/FTS *position* and *orientation* objects. This Max patcher takes a list of *positions*, projects these on the plane that best fits them, then rotates the plane (with projected *positions*) and eliminates z. This patcher enables subsequent fitting of 2-D parametric curves to the x-y coordinates (e.g. the x-y coordinates computed from the *positions* marking the thumb and index fingers).

an OpenInventor 3-D 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 specific 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.

- **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 3) 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..

- **Shape processing** - We have programmed *sheet*, a physical model of a sheet of two layers of masses connected through springs. The four corners of the sheet function as the control points and can be "attached" to e.g. the index and thumb tips of both hands.

- **Shape feature computation** - Many of the geometric computation objects and some of the hand feature objects can be used to compute shape features. Other type of shape feature computations involve superquadrics, a mathematical method to describe a wide variety of shapes with two parameters specifically related to shape (vertical and horizontal "squareness") and 9 others for size, orientation and position of the virtual object. Based on [9] we have programmed a *superquadric* object that fits a shape 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 shape described in terms of ellipsoid parameters (i.e. three size parameters) to the list of *positions* and computes within real-time. Currently our 3-D fitting and shape algorithms do not take into account that the virtual object should be bounded by the hands, so that the virtual object intersects with the hands. Thus, we need to add more constraints. For 2-D curve fitting we have pro-

grammed a *polynomial* object, which will approximate a list of *positions* by a polynomial of arbitrary degree. We are further studying 2-D and 3-D Fourier transforms for use as shape features [1].

4 Mapping Experiments

The main issue in mapping virtual object attributes to sound parameters is the choice of a suitable metaphor that reduces the cognitive load, or in other words "is easy to understand", for a novice user of the system. A mapping based on the real, physical world is 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 and virtual object volume to loudness. Informal evaluation showed this mapping to work well, i.e. it took only a few moments to get used to it. 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. Thus, the pitch of the sound was either fixed at one frequency, or was pre-programmed as a sequence. Instead the height was mapped to duration. The remaining available sound parameters in our simplified FM synthesis model were timbre specific parameters attack/release time, modulation index, high-pass filter amplitude and frequency and low-pass filter amplitude and frequency.

Given our development environment, we are debating whether the hands should always be "touching" the virtual object or whether the sound sculptor can "touch" at will and thus choose when to edit the sound and when not. In the latter case the difficulty arises how to give the sound sculptor effective feedback of the contact between his or her hands and the virtual object. We are experimenting with visual feedback.

The computation time for the mapping turned out to be a limiting factor for some of our experiments. We circumvented this problem partially by using the *client* object to communicate via a TCP message server between two Max/FTS applications, where one Max/FTS application would only run sound synthesis related patchers to obtain a high quality sound, and the other all the remaining patchers.

We furthermore found that when mapping a shape feature to a sound parameter, offsetting and scaling the value of the shape feature required some arbitrary heuristics based on workspace dimensions etc.. Although it is not a major problem, a solution without such heuristics would be preferred.

In our pilot studies, we have experimented with essentially two types of mapping, both inspired on the pragmatics of claying and both controlling sounds created by frequency modulation (FM) synthesis.

4.1 The Superquadric

The first experiment involved the manipulation of a virtual object with a shape that could vary between a sphere and a cube. The objects *ellipsoid* and *superquadric* were used to compute shape features. Manipulation of the virtual object occurred through the movement of any of the *voxels* that make up a hand. The *positions* of any of these *voxels* were taken as the *voxel positions* of the surface of the virtual object. Contact between virtual object and hands was continuous, i.e. the sound sculptor could not let go of the virtual object. The length of the main, i.e. longest, axis was mapped to low-pass filter amplitude and the transverse surface area was mapped to high-pass filter amplitude. Using the *ellipsoid* object, informal evaluation showed this mapping to work well. "Roundness" in one plane was mapped to the modulation index, whereas "roundness" in the transverse plane was mapped to attack/release time.

Although, as earlier noted, the *superquadric* object could not compute in real-time, the main problem with this mapping is that it is unclear how the sound sculptor intends to shape 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, the fitting algorithm searches for the best fitting superquadric for each new set of points without taking the previously fitted shape into account. This resulted in "shape jitter", i.e. the shape would at times jump from cube-like to ellipsoid without any significant hand motion, resulting in unpredictable and large sound variations. Possibly this can be circumvented by algorithm adaptations.

Nevertheless, we intend to explore this approach further because of the availability of the abstract shape feature "roundness", which may map well to parameters of similar level of abstraction such as "brightness". Brightness is used as a parameter in a perceptual sound model [10].

4.2 The Rubber Sheet

The second experiment involved 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. Contact between virtual object and hands was continuous. The computations could be completed near to real-time if a suitably low number of virtual masses was chosen.

Average length of the sheet, along the axis between left and right hand was mapped to low-pass filter amplitude and average width (i.e. axis between index and thumb) of the sheet was mapped to high-pass filter amplitude. This mapping is essentially the same as in the first experiment and worked equally well. A measure of the average planar curvature along the left-right axis was mapped to the modulation index and a measure of the transversal curvature along the same axis was mapped to attack/release time. This mapping was more difficult to use due to the "curvature crossover", i.e. when bending transversally the sheet would also bend planar occasionally, without the sound sculptor intending it. Thus, we need to define shape features more carefully, taking into account manipulation pragmatics.

5 Summary and Conclusions

We have initiated the development of sound sculpting, an environment in which a sound artist or musician can create, edit or perform sounds by changing attributes, like position, orientation and shape of a virtual object as input device, that can only be perceived through its visual and acoustic representations. We have identified three different types of sculpting methods that are regularly employed by humans and have discussed one of these for use in physical and abstract mapping strategies to relate virtual object manipulations to sound variations. The computation and use of other than the physical level of abstraction for the representation of gestural expression will exploit the capability of human gestures to effortlessly vary many degrees of freedom simultaneously at various levels of abstraction.

To implement these mapping strategies, we have created new software objects for our Max/FTS-based experimental environment to interface serial peripheral devices such as sensors, to compute geometric and kinematic parameters and to compute features of human body parts as well as virtual object attributes. To date, we have applied these Max/FTS objects in two experiments, one with a sphere-like virtual object and another with a sheet-like virtual object as input devices both for the control of FM synthesis parameters. We provided both auditory and visual representations of the controls to the user, with the aim to create an understandable and intuitive relation between the various representations. The main problems encountered were the search for an interaction metaphor that can quickly be understood, the incorporation of manipulation pragmatics in the mapping, "touching" of the virtual object and computability within real-time.

We believe that these approaches will lead to the reduction of the cognitive load in simultaneous multidimensional control tasks such as sound design. Further work is necessary to define useful operators that modify attributes of the virtual object and to define virtual object attributes that can be mapped to sound parameters. We hope to show through experimentation that virtual object attributes can be meaningfully mapped to sound parameters.

References

[1] Ch. Brechbuhler, G. Gerig, and O. Kuhler, "Parametrization of closed surfaces for 3-D shape description," *Computer Vision and Image Understanding*, Vol. 61, No. 2, March, pp. 154-170, 1995.

[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. 8, No. 5, pp. 977-984, 1997.

[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] Axel G. E. Mulder, S. Sidney Fels and Kenji Mase "Empty-handed Gesture Analysis in Max/FTS," *Proceedings of the AIMI international workshop on kansei - the technology of emotion* (Genova, Italy, 3-4 october 1997), A. Camurri (ed.), pp 87-90, 1997.

[5] 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.

[6] 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://fas.sfu.ca/cs/people/ResearchStaff/amulder/personal/vmi/HCI-gestures.htm

[7] 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://fas.sfu.ca/cs/people/ResearchStaff/amulder/personal/vmi/HMTT.pub.html

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

[9] 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.

[10] David Wessel, "Timbre space as a musical control structure," *Computer music journal*, Vol. 3, No. 2, pp. 45-52, 1979.