A VIRTUAL MUSICAL INSTRUMENT EXHIBIT FOR A SCIENCE CENTRE

A THESIS
SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE IN COMPUTER SCIENCE
IN THE
UNIVERSITY OF CANTERBURY
BY

GEOFFREY THOMAS

University of Canterbury
1995

To all my family, friends, staff members of the department, and everyone else who has been supportive and helped me in the production of this document. You know who you are!

Abstract

Virtual reality is a technology rapidly gaining interest from research and commercial groups around the world, but it's introduction into New Zealand has been slow. The majority of the general public have no concept of virtual reality, and only a few research institutes have begun virtual reality programmes of any sort.

Partially this is due to the high cost of 'off the shelf' virtual reality systems, which is usually beyond the range of many organisations. Also the complexity of the software and the knowledge required to create and manipulate this software makes it a daunting prospect for many.

This work describes the development of an economical system for the demonstration of virtual reality and some of its concepts and applications to the general public, in the form of an educational science centre exhibit. The system creates virtual musical instruments, overlayed onto the real world, and the user experiences these instruments as if they were in physical existence.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

> The primary defining characteristic of VR is inclusion; being surrounded by an environment. VR places the participant inside information.
>
> - Dr. William Bricken, HIT Lab (in [PT92])

> Now, with virtual reality, the window of the GUI that has kept us outside the screen looking in has dissolved, and we can step through the glass—replacing the desktop metaphor with a complete environment.
>
> - Pimentel and Teixeira (in [PT92])

Imagine being able to walk through a building before it was constructed, with the ability to move walls, doors, lights, furniture and other objects around at will, simply by grasping, carrying, and placing them, while all the time a cost figure is displayed at the corner of your vision. Or being able to have a game of squash with someone 300 kilometres away, simply by wearing a special pair of gloves and display system, and calling them up on a device similar to a telephone.

This is the concept of "Virtual Reality" (VR) — immersing a user in an artificially created environment, causing them to perceive it as a physically real environment.

Virtual reality is a relatively new medium to the general public, and it is likely that children, and even adults will have never experienced this new method of communication. A vehicle for conveying information about such a medium exists in the educational science centre, a place where the general public can learn about various topics in a controlled way.

1.1 What is "Virtual Reality"?

Virtual reality is an interface becoming popular all over the world as the latest form of human-computer interaction. The concept has been around for many years, but until recently, computing power and display systems have been unsuitable to make virtual reality a widespread and convincing interface. Its primary advantage over other forms of interface is that essentially there is no need to learn about the interface itself; the interface is transparent to the user. In many systems currently available the user of the system needs to learn how to manipulate the interface, before they can begin to use the system itself. In a virtual reality system however, the user acts directly upon the system, essentially bypassing any noticeable form of interface.

It has been sugested that virtual reality will be the television of the future, where participants can enter worlds of their own creation, and manipulate them in ways not possible currently. The promise of VR and its capabilities suggests it will be as pervasive as television, due to the imaginative freedom it will give to the participants. This will probably happen before the end of century, yet the majority of the general public seem unaware of virtual reality and its capabilities.

While virtual reality is becoming a well-known term among researchers, it has not had a great deal of impact on the general public. The average layperson is unaware of the technology or capabilities of this new medium. There are a number of reasons for this, such as geographical location and socio-economic climate, but the main factors for the slow infusion of virtual reality awareness into the general public, and indeed even research institutions, are technical and financial.

While the concept of virtual reality has been in existence for many years, only recently have sufficient technological advances been made to allow semi-mass production of VR systems. These systems, being practically the first of their kind, are expensive, as research costs need to be recovered by the manufacturers and production volumes are low. Added to that is the level of technical expertise required in manufacture, as VR is a high-technology industry, relying heavily on computers and other similar systems. Most VR systems are designed primarily for laboratory use, and therefore are not widely available for the public to experience. The software side of virtual reality is in its infancy, with researchers concentrating on hardware issues such as frame rates of display systems, and degrees of freedom of tracking devices. As a result, few systems are available that are oriented towards education of the general public about virtual reality and its promise as a new communications medium.

New Zealand is a country that tends to lag behind the rest of the world in high technology research, and virtual reality is one of the fields of technology to suffer from this isolation [Har93]. With the growing interest in educational science centres that are appearing in many of the major cities in New Zealand, an opportunity is available to increase the general public's awareness in this new and exciting technology. A possible virtual reality system for use in a science centre is described by Hart [Har91b], where visitors to the centre are taught about real life concepts and principles using a virtual reality system. The system Hart describes is hypothetical and uses virtual reality as a tool for teaching about other concepts. However, a virtual reality system can be used a stage earlier than what Hart describes, to teach science centre visitors about VR itself by giving them first-hand experience.

1.2 A Science Centre Exhibit

The educational science centre is a learning environment where concepts of practically any nature can be taught to visitors. While they are mostly aimed at school-aged groups, adults also form a significant percentage of science centre visitors. The biggest attractive aspect of the science centre is interaction; exhibits are designed to convey some principle by allowing users to experience the principle, or at least enough parts of it to grasp the underlying concept. Static displays, while present, are a minor number of the exhibits.

An example of a high technology-based science centre exhibit is the giant working microprocessor at Silicon Valley's Tech Garage. The exhibit shows data moving into, out of, and around the CPU with lights showing the flow of electrons. Another example is the computerised "Game of Life" display at the San Francisco Exploratorium. Most of these are implemented in a low technology way, with creative use of lights and sounds to convey the idea of the exhibit.

Some aspects of exhibit design (from [AS84]) are covered in Tables 1.1 and 1.2. Because of the potentially large number of exhibits that will be present in a science centre environment, the time and cost of maintenance of exhibits needs to be kept low. With a large number of visitors using the exhibits and the high hands-on nature of the exhibits, hygiene is a major concern. Interactivity is the major aspect that sets science centres apart from museums, and this should not be underutilised in the design process. Depending on the funding scheme employed by the science centre, finances for exhibits may be scarce, and costs should be low. However, the cost of an exhibit needs to be considered for the long term as well as the short term, — a $10,000 hologram with no maintenance costs will prove to be less expensive in the long term than a $2,000 exhibit that costs $600

Strongly negative	**Neither negative or positive**	**Strongly positive**
Badly placed; not easily noticed	Participatory	Makes topic come to life
Does not give enough information	Deals with topic better than a textbook	Makes its point quickly
Attention is distracted by other displays	Artistic	Has something for all ages
Exhibit is confusing	Makes a difficult subject easier	Memorable

Table 1.1: Visitor perceptions about exhibits (from [AS84])

every month. The nature of the exhibit is important aspect and needs to be considered carefully, especially in technology based exhibits. Bell [Bel92] points out that computer based displays are generally given a poor reception by visitors to science centres because computer systems have advanced greatly over the last few years, and immensely over the last few decades. This is basically true of technology displays in general, because of the rapidly changing nature of technology. In contrast to this, basic physical principles such as gravity, air pressure, and magnetism haven't changed since they were discovered, and so exhibits for these, while potentially old, are not obsolete. A virtual reality exhibit is a technology-based exhibit, and so the application running on the exhibit must be carefully chosen as the case could arise where, for example, video arcade games in the lobby were more interesting and technologically advanced than the exhibit on the exhibit floor.

This thesis covers the development of a virtual reality exhibit for the science centre *Science Alive!* in Christchurch, New Zealand. Factors concerning design and functioning of the exhibit are covered and solutions are presented for many of them.

Undesirable Characteristics	**Desirable Characteristics**
Topic not sufficiently explained.	It's obvious where one should begin and how one should continue.
The exhibits are not realistic enough; difficult to relate to the real world	Uses a lot of modern display techniques which help one learn.
It is appealing to children but not so to adults	It uses familiar things and experiences to make its points.
It is traditional in style; old fashioned.	It includes a comprehensive display of objects and/or specimens.

Table 1.2: Desirable and undesirable characteristics of exhibits (from [AS84])

Chapter 2

Virtual Reality

Over the last few years a way to interact with computers has become available that offers a new way for humans to communicate with machines and with each other. This method differs from the current widely used interface — two dimensional graphical user interfaces (GUIs) — and provides an interface that immerses the user within their data, thus providing a three dimensional experience through the computer. The new interface is known as "Virtual Reality" or VR, and will play a major part in computing and society in the coming years.

2.1 "Yet another human computer interface!"

> The key goals in user interface design are increase in speed of learning, and in speed of use, reduction of error rate, encouragement of rapid recall of how to use the interface, and increase in attractiveness to potential users and buyers.
>
> *From: Computer Graphics: Principles and Practice* [FvFH90]

The methods of using computers have progressed greatly since they first came in to use, from switchboards, punched cards, keyboards and visual textual display units, to mice and GUIs. Because of recent advances in computing hardware, and requirements by some applications for 'hands-on' interaction with large data sets, another interface technology has been developed.

In this technology, two factors predominate — increased visual emphasis, and greater ease in the learning of applications. Human-computer interfaces have been progressing further away from the computer's 'language' and more towards a human-oriented system, of 'natural' and intuitive commands which utilise a high visual aspect. Such interfaces require much less learning time than their predecessors, mainly because of the relationship between real world items and events, and items and events used in the interface to achieve similar functions. An example of this is the "Trash can", used in the Apple Macintosh GUI for deleting files; users 'throw away' files as they would a piece of rubbish.

Interface interaction has been simplified resulting in more 'user-friendly' computers. Users do not need to learn obscure commands with many options, such as those in the Unix operating system. An example of the above evolution of user interfaces is the difference in actions required to perform some desired result. In a command-line based Unix system, the commands:

```
cd XXX<RET>
ls -al<RET>
```

produce a listing of the contents of the directory *XXX*. The equivalent result on a GUI-based Macintosh computer is achieved by double clicking on the folder icon named "XXX", which opens folder XXX and displays its contents. The trend in interfaces is a lessening of knowledge required to use the interface itself to achieve a desired goal. Actions are easier to learn and more intuitive to new users.

The general trend of lowering the user's knowledge requirement is also applicable to virtual reality interfaces where the interface accepts 'natural' human gestures as commands.

VR is an example of what Schneiderman refers to as a *direct manipulation interface* [Shn87], whereby the user acts directly on an object. Manipulation of objects is directly related to the manipulation being done, as is the Trash can case mentioned previously with Apple Macintosh computers. In virtual reality, the user could grasp a virtual object, and drag it to a virtual trash can for deletion. The user has direct control over the data, rather than an abstracted link as is the case with command-line interfaces. For example reaching out and grasping at something in a virtual environment generally means "pick that up". It can vary however from one implementation to another; VR interfaces are in an infancy state and there is no common standard that is applied; contrast this with the current selection of GUIs, that all have major features in common, such as icons, dragging of objects, "go-away" buttons. A study of VR as a communications medium is presented in [Har91a].

Data visualisation is one of the areas that is benefiting from improvements in interface design. Due to major advances in visual display technology, interest in computer aided

I		II		III		IV	
X	Y	X	Y	X	Y	X	Y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

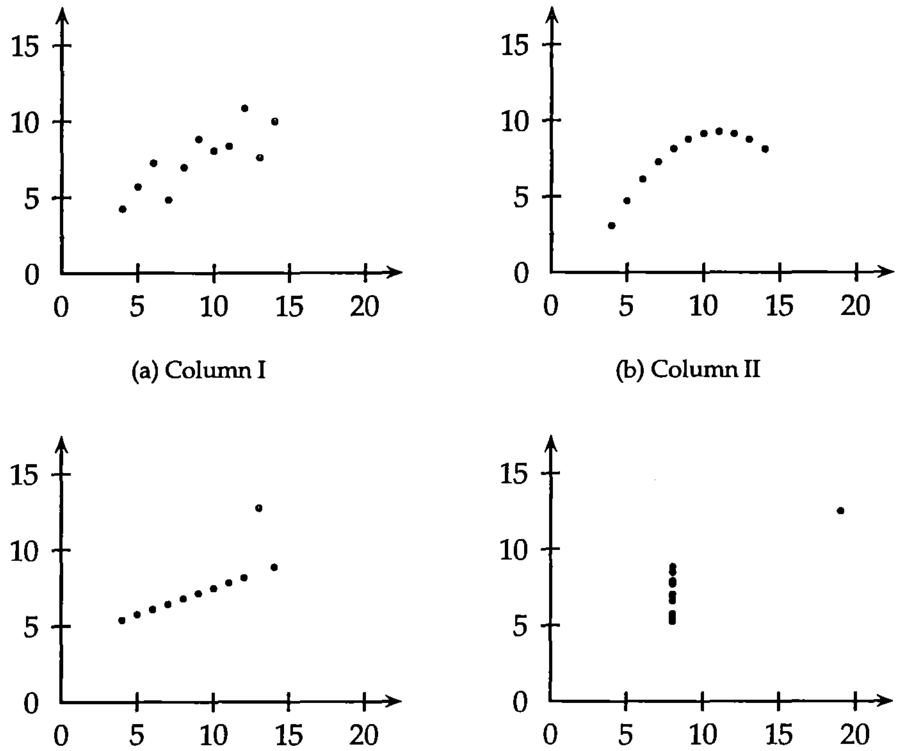
Table 2.1: A selection of data values, with statistically similar properties (from [Tuf86])

data visualisation has increased rapidly over the last ten years. It is now reasonably straightforward to produce intuitive and informative complex graphs of abstract data models, whereas a decade or two ago such displays would have been considered almost solely in the realm of science fiction.

Statistical measure:		Value:
N	=	11
mean of X's	=	9.0
mean of Y's	=	7.5
equation of regression line: Y	=	$3 + 0.5X$
standard error of estimate of slope	=	0.118
t	=	4.24
sum of squares $X - \bar{X}$	=	110.0
regression sum of squares	=	27.50
residual sum of squares of Y	=	13.75
correlation coefficient	=	.82
r^2	=	.67

Table 2.2: Statistical attributes of the data in Table 2.1

Table 2.1 shows the data from a group of data series, each with statistically similar features. In Figure 2.1 the graphical representation of the data is displayed, showing relationships that are not immediately obvious from looking at the list of data points. Observing relationships in datasets that contain, for example spatial information, becomes much easier if represented in a spatial way such as Figure 2.1, rather than in a 'raw data'



(a) Column I

(b) Column II

(c) Column III

(d) Column IV

Figure 2.1: Graphical display of data in Table 2.1 (from [Tuf86])

format such as in Table 2.1.

This is one example of how different forms of data visualisation can show different features of the data, and that these features are not obvious when using one particular visualisation method. As a result of human-computer interface enhancement, data visualisation users have been able to visualise increasingly complex systems, and the interactive nature of current interfaces makes such visualisation a simple task. Parameters to complex simulations can be changed and the results seen immediately on a graphical display. Using a text based system to achieve the same results is often much more difficult and time consuming.

A further step in visualising data is now available using VR technology. Users can now create artificial environments in which to view and manipulate their data in a natural intuitive way, changing features or parameters of the data at the wave of a hand, or altering the behaviour or structure of a simulation for example, by 'moving' parts of it around.

Current methods for viewing and manipulating these environments are predominantly carried out with two dimensional interfaces (GUIs), but the required tools to view such data in three dimensions are now becoming available, with the help of VR technology.

Examples of such environments might be the architecture walk-through mentioned in Chapter 1, a simulation of a large data set or complex system such as a storm cloud, chemical molecule or wind tunnel air streams, or an entertainment setting, such as an interactive, immersive version of "Dungeons and Dragons".

In its very basic form, virtual reality is the next generation human-computer interface [BM91, Fis90, Fol87].

2.2 Description

To describe virtual reality, it would seem obvious that a definition of reality itself is required. What is reality? This is a difficult question to answer, and very subjective. Therefore, rather than describe what reality is, it may be more appropriate to describe some generally agreed upon features of reality.

2.2.1 Consistency of sensory input

Sensory information conflicting with our other senses or with our internal 'model' of reality disrupts our perception of something being real. The behaviour of things should match our real-world expectations. From past experience in reality, we expect real world events to have predictable results. For example, our day-to-day experience of a large number of real-world stimuli about dogs suggest that dogs in general can bark and do not 'moo' like a cow. If we hear a barking sound, and see a dog going through the motions of barking, we associate the sound with the dog. If we hear a mooing sound, but still see a dog going through barking motions, (and perceive that the dog made this sound) then our sense of the dog being real tends to be stretched.

Aukstakalnis and Blatner [AB92] offer this example:

> If we hear a dog in front of us and then *see* a dog in front of us, not only do we tend to think that the dog we see is the dog that made the sound, but also that the dog and the sound are 'real'. Neither of these may be true.

This conflict of sensory inputs not only affects our perception of reality — it can have negative psychological effects. The visual sense is of particular importance. While it can 'override' the other senses at a conscious level, at a subconscious level it is not as dominant and can result in 'Visually Induced Motion Sickness' (VIMS), or 'Simulator Sickness' [HR92, AB92, Bio92].

VIMS can result in three main categories of discomfort — nausea, dis-orientation, and occulomotor problems such as eyestrain [LKL+92]. Some symptoms are involved in more than one category, for example blurred vision fits into both the occulomotor and disorientation categories.

The vestibular system — the set of tubes located in the inner ear that informs us of movement and rotation — seems to be the main cause of the VIMS effect, contradicting the visual information we are receiving. An example of this is a passenger travelling in the back seat of a car without looking out the windows. The passenger's visual system informs them they are not moving relative to most objects in their immediate environment because a large proportion of their visual field shows no movement. However their vestibular system informs them of movement. They are in motion while apparently stationary. The reverse situation also gives rise to VIMS — being stationary while apparently in motion — a situation applicable to many virtual reality displays. It is interesting that subjects with defective vestibular systems do not experience motion sickness [Ebe92, MS92]. McCauley and Sharkey [MS92] note that in driving simulations, older subjects were more susceptible to simulator sickness. This conflicts with Biocca's reports [Bio92], where reports of motion sickness in subjects over 50 are few. Noticeable time lags between head movements and display updates when using head mounted displays in VR can also cause VIMS. Inertial effects on our limbs when travelling in a car (or not travelling in a virtual environment) aid the vestibular system in detecting motion (or lack of it) in our bodies.

Another factor is the position of the subject; the driver in a car generally doesn't suffer from motion sickness, unlike the passengers [PCC92]. Similarly the person controlling a simulation often suffers less from the effects of the simulator than the other participants.

Simulator sickness can also be caused when results of motions in the real world differ from results of motion while in the virtual world. For example, turning 30° to the left, should be matched by a 30° change in view in the virtual environment. In the particular case when there is a delay in movements of the head, and the display in the visual field of the user being updated to reflect the new viewpoint, simulator sickness can occur depending on the severity of the difference [HR92].

The effect of VIMS can be overcome to a certain extent. Participants can adapt to small 'defects' in simulators and VR systems, and the effect of adjustment can linger after the experience as the human system re-adjusts to the real world. The effect also appears to be lessened if users are introduced slowly to the virtual world; subjects adjust to the effects of VIMS and the effects are lessened on subsequent visits. Such effects caused by the simulators are far less frequent by the sixth visit, and after a number of months' abstinence from the simulator subjects who had a number of previous experiences readjusted quickly to the simulator again [Bio92].

The body's priopreceptive system, which informs us of angles our limbs are to each other, by the tension or lack of it on our muscular and skeletal systems, also plays a major part in virtual reality. If we move an arm upwards, tensions in the muscles required to hold it there inform us that the arm is not lying limp at our sides. This effect occurs even without visual confirmation that our arm has moved. Consider then what happens when we move our arm above our head, but the virtual reality system we are using does not show this, or shows that we have only moved our arm perpendicular to our bodies. Due to the discrepancy between the priopreceptive and visual senses, we will probably attempt to compensate the visual discrepancy by moving our arm further, but be puzzled because we can already feel that our arm is held up.

The earlier example of travelling in a car also affects the priopreceptive system, by way of the inertia experienced by our limbs. If we are seated in a car, (or in a more extreme case, a jet aircraft) and the car accelerates rapidly, inertial effects push us into our seat. This is sensed by a number of our senses (priopreceptive and touch).

Currently in virtual environments, consistency between sensory stimuli is not fully utilised due to mechanical difficulties and the lack of understanding about the inter-relationships between the sensory systems. Many VR systems have only one output system, that of the visual display. Some also have aural output, and a few have various tactile and haptic feedback systems.

2.2.2 Real-world correspondence / Display mismatch

Issues of time and space also affect the correspondence between the virtual and real worlds. We expect certain results from certain actions to occur within a particular time frame, especially with natural laws which we have experienced and had reinforced since conception. If we stop suddenly in a moving car, we expect to be thrown forwards by inertia. If we look at our hand, and move it while we are watching it, we see the movements immediately. We expect a response to certain actions to take place within a certain length of time – any longer and it begins to appear unrelated to the original action (see the case cited in Section 2.2.1, where there is a delay in movements of the head and the update of the display in the visual field).

Aukstakalnis (ibid) states that to feel immersed in a virtual world, we must be surrounded by " various stimuli in a manner that makes sense and that follows rules similar to those of the real world. That is, when you turn your head to the left, you see the objects to the left of you. When you walk forwards, you get closer to the objects in front of you. These are elementary features of our sense of being immersed in an environment; and when you're in a virtual environment you expect the same results." [AB92].

2.3 History

The concept of virtual reality, involving users in an artificially created environment, is relatively old, with its beginnings in the movie and entertainment industries. Stereoscopic movies in the early 1950s, and wide-angle projection technologies such as Cinerama and Omnimax were immersing entertainment seekers long before the term virtual reality came into common use [Fis90, Rhe91].

These technologies are not interactive: the viewers could not control their viewpoint, and only visual and sound cues were presented, but other systems such as the Aspen Movie Map allowed the user to control their viewpoint and the Sensorama arcade game designed by Morton Heilig in the 1960s provided motion sensations and aromas to the user [Fis90]. Computer generated virtual environments were pioneered at MIT in the 1960s, by Ivan Sutherland. Sutherland foresaw the possibilities of virtual environments in his discussion on "The Ultimate Display" [Sut65]. He went on to build a visual display that had many of the features of present day virtual environment displays; a stereoscopic view, coupled to a tracking device and capable of altering the views presented to the user in real time, depending on what direction they were facing.

Recent technology advancements, notably the liquid crystal display (LCD), and improvements in computing power, have made complex virtual environments relatively commonplace. Computing power and graphical rendering can now display shaded, illuminated and complex virtual environments, containing many objects; a far cry from the simple wire-frame images presented on Sutherland's display. Spin-off areas of research have also been created from the development of virtual environment technology.

Current work at NASA Ames, one of the better known virtual reality laboratories, involves the Virtual Environment Workstation, a system of liquid-crystal stereoscopic displays, 3-dimensional stereo sound, and tactile feedback for the gesture-controlled input [Fis90, Fis91, Fol87] .

Other major institutions working in the virtual reality field include the University of North Carolina at Chapel Hill, where new chemicals are being designed in a virtual environment system by picking up molecules and fitting them together like Lego blocks, and the Human Interface Technology Laboratory at the University of Washington, where research is being made into high resolution displays and other advanced technology products [Rhe91].

2.4 Telepresence

Typically virtual environments are entirely computer generated, although in some cases they consist of a computer augmented view of a real world scene. Removing the computer generated aspect from the display, and replacing it with a video signal from a camera (or pair of cameras for stereoscopic display), is the principle behind "telepresence", where remote sensory inputs such as those received by video cameras and microphones mounted on a robot can be experienced by a user at a different location. The user is given a sufficient degree of sensory feedback to perceive being present at the remote location. For example, a robot outside a space station can be manipulated by an astronaut inside the station, who perceives that they are performing the task that the robot is doing, without physically being there [BM91, Fis90]. At the Naval Ocean Systems Center (NOSC) in Hawaii, a telerobotic exoskeleton called the "Green Man" has been developed that gives the wearer of the exoskeleton control over, a view and audio feedback from a remote robot system [Utt89, Rhe91, AB92]. Another telerobotic system in use at NOSC is the High-Mobility Multipurpose Wheeled Vehicle (HMMWV), a jeep fitted with a pair of television cameras providing stereoscopic vision to the operator, audio microphones and speakers to allow two-way aural communication, and control systems to direct and control the jeep itself.

Simulator sickness (see Section 2.2.1) is one of the major obstacles in telepresence applications, as by their nature, many such applications require the remote robot and sensor system to be in motion, while the operator is usually stationary. The operator is therefore presented with conflicting motion cues, giving rise to simulator sickness.

Teleconferencing is an aspect of telepresence that is interesting many communication providers around the world as a new method of running conferences. In cases where people who are spatially distant from each other wish to communicate, telepresence can give them the feeling of being together in the same room without being physically present. At a set time, the spatially separated participants of the teleconference don their VR equipment at their respective locations, and set up a connection between themselves and the other participants.

Each participant is shown a view from around a conference table for example, with the other participants present also around the table. The conference proceeds as if the participants were physically in the same room, at the same table. Benefits of this approach over the physical method of actually being seated around the table are that travel to the conference site is avoided, and time and cost is not incurred as a result. If the participants are from all around the globe, and some are in remote locations, the time and cost of travel could be high enough to warrant not having the conference. Teleconferencing also provides a vehicle for other computer based presentation methods, such as multimedia. Aukstakalnis (ibid) gives the example of a graph or report being presented floating in space in front of the conference attendees [AB92].

One telepresence application has been recently used in education centres in the United States. Known as the Jason project, a group of science centres was in communication with a deep-sea exploration team via satellite and other real-time communication mediums [Tyr89, Bal92]. The exploration team used video cameras mounted on remotely operated vehicles (ROVs) to relay live images to the science centres where they were viewed by students on various television displays. The students were able to talk to researchers aboard the vessel from which the ROVs were being controlled, and ask questions of the researchers as the ROVs were being maneuvered around sunken wrecks and other interesting sea-bed phenomena.

2.5 Presence in virtual environments

The main aspect of virtual environments is the feeling of immersion; being somewhere 'else'. To feel present in a virtual or remote environment, we have to feel immersed in that environment. Such immersive experiences already surround us everyday, in the form of books, television and movies for example. Aukstakalnis (ibid) suggests three levels of immersive experience [AB92]:

Passive : The user has no ability to change the environment; there is no interaction with objects in the environment, and they have a viewpoint not under their control, but under the control of some other entity. Such a system is typically only output; there are no input devices connected to it.

Exploratory : There is no interaction, as in the passive case. However, the user's viewpoint is under their control; they have the ability to move the viewpoint through the environment directing what they wish to see, but have no ability to manipulate the objects present.

Interactive : The user has the ability to alter the environment and their viewpoint. This is the highest level of immersive experience.

Currently there is as yet no definitive method defined to measure presence [She92, HD92], although three possible independent 'variables' of presence are suggested by Sheridan [She92]:

- level of sensory information;
- ability to change sensory 'viewpoint' (this could be aural, haptic, or others, as well as visual);
- ability to change the environment.

Heeter [Hee92] gives a different three variables of presence. *Subjective personal presence* is a measure of the extent to which and the reasons why you feel you are in a virtual world; *social presence* is the extent to which other beings, living or synthetic, also exist in the world and appear to react to you; em environmental presence is the extent to which the environment itself appears to know that you are there and reacts to you.

While no standard form of measuring presence is defined, it is possible to compare experiences to determine if one felt more immersed in experience A than in experience B. This allows the system to be enhanced to increase the level of presence, while only making a rough estimate of which aspect of presence is being changed.

2.6 Applications

Following is a small sample of some of the possible and actual applications of VR, taken mostly from [AB92]. In a surgical simulation, a surgeon practices on a virtual cadaver, planning an operation, and then performs it in the real operating theatre. A radiation therapist can plan the direction of radiation using virtual radiation beams on a represenation of the patient, to minimise chances of hitting healthy tissue (existing UNC application of this [AB92]). Immersive games give players a sense of actually being there. In interactive theatre a number of participants create a unique experience by interacting with others in some rendition of, say, a Shakespearean play. Playing a multiplayer sport with a number of other participants from around the town, country or world, in the same virtual space is another example of the possible uses of virtual reality.

While VR has applications in almost every aspect of human life, the applications most likely to see the technology first are those that benefit mostly in terms of entertainment value and monetary gain. However, many 'small' research applications — ones not destined for commercial use in the short term future — exist in various laboratories around the world. UNC used VR to design their new Computer Science building, before actually building it. They discovered while using the architect's drawings as a basis for a virtual world that an unforeseen problem existed with the lighting. The architect was not convinced until he also ventured into the virtual model, and saw the 'defect' for himself.

Large datasets in many degrees of freedom can be more intuitively displayed using VR technology. A UNC system exists to manipulate molecular models, to determine how new proteins and compounds will behave prior to synthesising the proteins.

Huge amounts of financial data, can be visualised quickly and trends can be seen more readily via VR, as each important degree of freedom of the data can be mapped to some aspect of the VR environment (time to X axis, amount to Y axis, change to colour; sounds and tactile feedback can also be included).

Education is one of the areas most likely to benefit from VR. The concept of virtual reality offers possibilities of learning experiences that cannot be obtained in conventional teaching methods. Christel [Chr92] points out that when applied to training and education, virtual worlds offer the potential to teach in ways that would otherwise be too hazardous, costly or impossible. He also notes that time constraints make performance of some real-life training tasks difficult, whereas these time constraints can be altered or removed with virtual reality systems. For example, consider the exploration of a thunderstorm, being able to move the viewpoint around through a volume of virtual air, observing changes in temperature and pressure as changes in colour. Then to be able to speed the entire system up tenfold and observe how the cloud changes over time. Or perhaps to venture into a virtual bee colony as a worker, and observe the behaviour of the bees in the colony and their reaction to this new member of the hive.

These examples could be taught with conventional teaching methods, but would be a third person approach, and not interactive.

Virtual reality technology can be used to explore and discover things about the world we live in, in a way that current learning methods cannot. Some examples of this are:

- the ability to view a molecule on an atomic scale, changing its structure and observing the result while still at an atomic scale;
- experiencing life in a race, culture or species that has long since disappeared from the earth, or is of a non-human nature, such as the ancient Babylonians, or a South American leaf-cutter ant colony;
- observing physical processes such as electrons moving in a wire, photons escaping from the sun, or the passage of a water molecule as it goes through the water cycle (rain, river, evaporation, etc.);
- travelling inside objects that are too small or fragile to be observed in such a way in reality for example a voyage through the human body as in the movie "The Incredible Journey", or through the interior of a flower.

Other examples can be found in [Pan93, ST91, FL92]. Some real educational applications have been constructed, such as a virtual physics laboratory [Yam93], the Creative Technologies Project (a testbed to analyse the spatial skills of children) [Mer90], and a juggling application by VPL's Jaron Lanier [Fri91].

Virtual reality gives the student a unique opportunity for insight into the function and form of the object(s) being studied, that may be impractical or impossible to perform in reality. Such methods of teaching provide a more "hands-on", "up-front" style of teaching than pen–and– paper methods. Recall the scientific visualisation examples in Section 2.6. Presenting a large array of numbers to a student and asking them to see the pattern present in the numbers and to understand its significance is a much more daunting task for them than being able to see and manipulate a multidimensional representation of the numbers in virtual reality, incorporating many of the student's senses.

Because of the powerful effect VR can potentially have on education it is likely to be very soon that VR systems will replace home computers in schools and other learning institutions.

Chapter 3

Components of a typical interactive VR system

The main feature of VR is its immersive quality, i.e. the placing of the user into a foreign, usually synthetic environment. This typically involves the use of a visual display system, that presents a controlled view of the environment to the user, and input devices of some sort, usually based on a glove type instrument that allows the computer to determine information about the user's hand, such as position, orientation, and the status of the fingers. A glove metaphor is the most common form of input device used as this allows 'natural' motions of the user's hand to be mapped to commands to the computer, making the interface more transparent than mouse-and-icon based 2 dimensional windowing systems. The hand is used most often in the real world for manipulating objects and is one of our most dextrous limbs [Sut65].

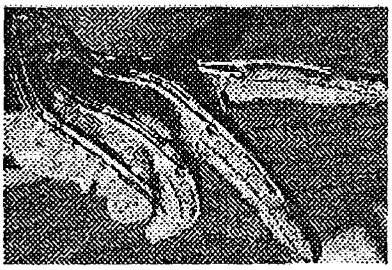
The quality of the display systems have a great influence on the quality of the perceived 'realness' of the environment, or the degree of immersion that the user experiences. Factors such as lag in tracking devices, rates of visual field updates and behaviour of objects in the virtual environment can drastically alter the degree of immersion. Such factors therefore need to be taken into account when developing a virtual reality system (see Section 2.5).

3.1 Input devices

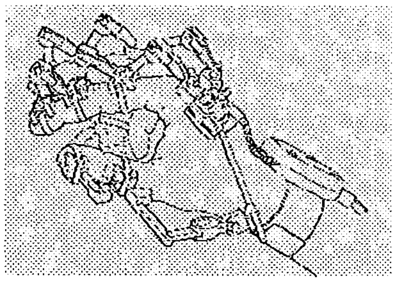
To control and interact in the virtual environment, some form of input device is needed. A number of systems are used in current VR systems for input of the user's commands. The most common input devices are used in the process of limb detection, as this is the primary form of interaction with the physical world.

Limb detection : The position, orientation, and motion of limbs needs to be detected. Just how much depends on the application; for the most part, hands are used for input, and gloves are used as the input device as they accurately report finger flexion and position, and position and orientation of the hand.

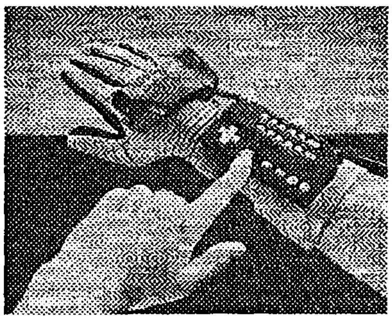
- Gloves : the glove metaphor has emerged as the most common VR interface due to the 'natural-ness' of using the hand as a manipulative tool. The hand is the primary limb on the body, with highly developed sensitivity to touch, temperature, and other contact-based phenomena experienced in the real world. Below is a listing of some glove devices commercially available at the time of writing (see Figure 2):



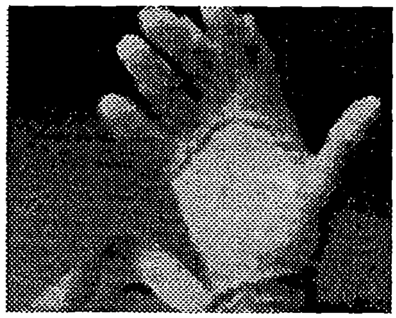
(a) VPL DataGlove



(b) EXOS Dextrous Hand Master



(c) Nintendo/Mattel PowerGlove



(d) Virtex CyberGlove

Figure 2: Various commercial glove devices

a) DataGlove: Manufactured by VPL. Has fibre optic strands in the fingers that determine finger flexion from the amount of light lost when bent, and electromagnetic tracking via Polhemus tracker. High cost laboratory instrument. Very accurate.

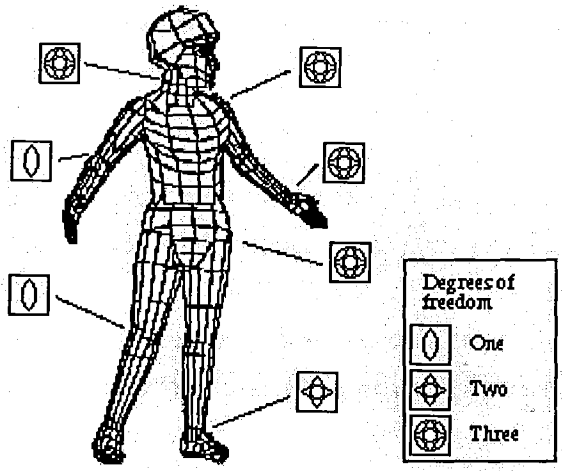
b) EXOS Dextrous Hand Master: uses hall effect sensors to measure finger bending, and choice of trackers (usually an electromagnetic one) for position and orientation. Very high cost, but extremely accurate.

c) PowerGlove: manufactured for the Nintendo Entertainment System (a home video game) [Egl90]. Uses resistive strain gauges in fingers to give degree of finger bend, and ultrasonic trackers to give position and 1 axis of rotation. Low cost and robust. Not very accurate or precise.

d) Virtex Cyberglove: similar to DataGlove in appearance, and PowerGlove in design. Highly accurate strain gauges in fingers and choice of tracking devices for usage.

e) Others used in specialist applications (e.g. teaching deaf/sign language/etc.)

(a) User wearing DataSuit



(b) Degrees of freedom sensed on limbs

Figure 3.2: DataSuit from VPL

- VPL's DataSuit: this is a wet suit fitted with sensors to measure the positions and angles of the user's limbs. Only a few datasuits have been made; the unit is not in commercial production, but is used as a testbed.
- Videoplace and friends: un-encumbered VR. The user's limb and finger status inferred from analysis of video images of them, in real time.

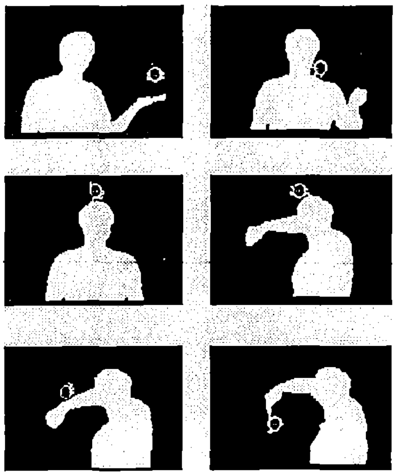
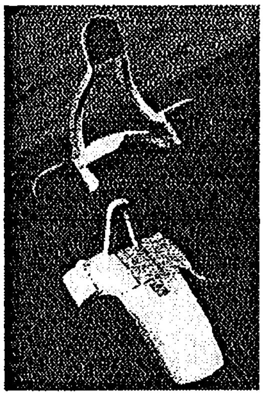


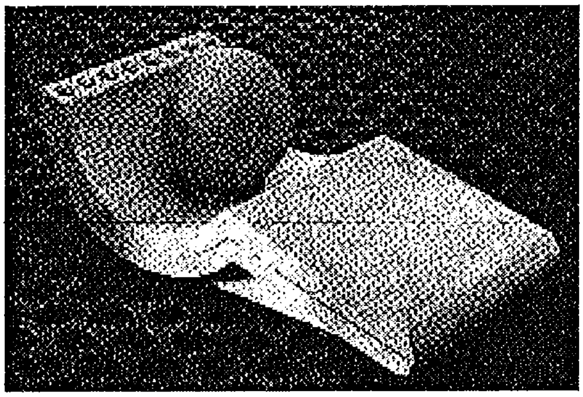
Figure 3.3: Myron Krueger's "Videoplace"

Other input systems : While limb detection is the most common form of input device for virtual environment systems, other systems are also in reasonably common use. As well as techniques that are being explored to give new methods of interaction with the virtual environment, some more commonplace devices are also used for input tasks.

- Facial expressions: work is in progress at the Advanced Telecommunications Research Institute International (ATR) [Rhe91] on recognition of facial expressions. ATR view the face as being one of the primary vehicles in human communications and have been developing non- encumbering devices for transmitting and displaying three dimensional facial images. The Waldo, described in [AB92], is used by animators to control an animated character's facial expressions.
- 6DoF mice: Logitech have recently released a mouse using ultrasonics that can be operated either as a conventional 2DoF mouse or a 6DoF head tracker.
- Spaceball: force sensitive ball that gives 6DoF input.
- Voice commands.
- Conventional mice: used on some systems that don't require complex input (e.g. Virtus Walkthrough).
- Joysticks: a reasonably intuitive interface.



(a) Logitech 2D/6D Mouse



(b) Spaceball

Figure 3.4: Some other input devices

The kind of input device most suited to an application depends on the task to be performed. A 'simple' device such as a Spaceball, joystick or mouse is well suited to navigation tasks where a simple movement metaphor is required, but may not be the most appropriate device when fine dextrous manipulation is required, such as insertion of a complex shape into another complex shape under various constraints, or manipulation of a complex object's structure.

3.2 Output devices

For a virtual environment to be of any use, it must display something to the user. The medium of display can be of any type that can be sensed by humans. Some of the more common ones are visual, aural or acoustic, and tactile and related senses.

Visual

There are 3 categories of visual display systems, which classify visual displays based on the display's handling of a 3 dimensional scene. *Stereoscopic*, or *binocular* displays feed a separate and slightly different image to each eye, emulating the binocular disparities or 'stereopsis' due to varying depths, that the human visual system experiences in reality. *Bi-ocular* displays feed the same image to both eyes, which reduces the computational load on the visual rendering system, but also removes the disparities present in the binocular displays and real life. *Monoscopic* or *monocular* displays only feed a single image to one eye. There is no depth information from stereopsis available in monoscopic systems. Two monoscopic systems may be able to be combined to form a bi-ocular, or binocular system, depending on the images rendered on each of the displays.

Display systems are also classified according to the method used to view the displayed image(s). *Head mounted displays* (HMDs) are those that are worn typically as a helmet, in a way that the user's head movements also move the HMD, and change its position and orientation. This makes it a 'hands-free' type of system, where the user does not have to keep moving the display when they wish to change their viewpoint. Such systems need to be light, manoeuvrable (low inertia, or small), and pleasant to use, as most users find that a heavy HMD, or one that has its weight distributed poorly (to the front, for example), is very uncomfortable and tiring to wear. Such displays are also known to be a cause of VIMS [DL92].

Head coupled displays are mounted on a separate mechanical linkage system that bears the weight of the display. This type of system is less physically demanding on the user, but it must be manually re-positioned and oriented, as it is not connected to the user's head. An advantage of using a head coupled display is that a better optical system can be used as weight and size are less of a consideration. For this reason, head coupled displays have typically used cathode ray tubes (CRTs) and have resolutions orders of magnitude higher than those found on HMDs, which are mostly based on liquid crystal displays (LCDs), because of the much lower weight of LCDs compared with CRTs.

Some VR applications use an ordinary video monitor as a display device. Such applications have lower demands computationally, only rendering a single monoscopic image. The processor time saved from this can be used to improve the environment simulation, giving more realistic behaviour to the objects being simulated, but less of an immersive feeling to the whole system (projection technologies can improve this; see [Ral94] for an example). Factors that can affect the level of immersion in the virtual world include:

- the display's *field of view*, where a 'wider' display (i.e. more peripheral vision) gives a greater sense of immersion;
- the display's *resolution*, the higher the better. However, currently many visual displays and certainly all LCD based HMDs are many orders of magnitude lower than real life: a human eye with 20/20 vision can resolve approximately 1 arc minute, or 60 'pixels' per degree (5400 pixels per 90°) [AB92];
- the display's *update rate*, which includes the time taken to generate a video frame;

- the display's *image quality*, where black and white images, or wireframe images give less of a sense of immersion, due to the difference between the real world (full colour) and the virtual world.

For displaying 3D objects in the real world, a number of displays have been manufactured based on the normal video monitor, but with the addition of a vibrating mirror, to give height information [HB86].

Some examples of the various types of displays are:

- normal video monitors;
- 3D volumetric displays using, for example, conventional CRTs and flexible mirrors [HB86];
- HMDs (e.g. Eyephones, UNC system) [AB92, Rhe91];
- MIT Holographic display that renders a three-dimensional holographic image in space in front of the user [VRN92];
- Head coupled displays (e.g. BOOM) [AB92, Rhe91];
- Laser Microscanner, that scans a modulated laser beam onto the retina, constructing the image[Rhe91];
- Private Eye, an array of light-emitting diodes (LEDs) that shine on a rotating mirror, producing an image of the screen [Pau91, AB92].

Aural, or acoustic

A sound's behaviour in a virtual environment should match its behaviour in the real world. Usually, the participant in a virtual environment is wearing headphones, receiving aural information from the computer. The sound is altered to a large degree as it passes into the eardrum by the mechanical construction of the ear-lobe. This causes the sound to vary from the original source, according to some function dependent on the direction from the listener to the source. This direction and listener-specific transformation of the sound is known as the *Head-Related Transfer Function*, or HRTF. To reproduce this effect in virtual environments, a number of systems have been constructed to accomplish the complex signal processing required. A simple solution is to have audio speakers surrounding the user for every possible position that sounds in the virtual environment could take, and to feed the sound through to whichever speaker is in the matching real world position (i.e.

same azimuth and elevation relative to the user's head). Other more complex (software-wise) solutions involve convolution of the audio signal. The Convolvotron, and the Virtual Audio Processing System both use digital signal processors to achieve the desired effect [AB92, Rhe91].

Tactile/Haptic/Priopreceptive

The tactile sense, or sense of touch and senses related to it provide a large amount of information about the physical world that our visual and aural senses may be incapable of detecting. For a blind or deaf person, their tactile sense becomes their 'eyes' or 'ears'. Even for a non-disabled person, tactile events have a strong impact on everyday existence in the real world. We are constantly 'feeling' things, from the brush of clothing moving on our arm, to the movement of air past our face and the impact of our feet with the ground as we walk. The tactile sense is most highly developed in our hands and face, with approximately "17 million mechanoreceptors throughout the hand" [AB92]. Texture, temperature and shape can all be garnered from an object via tactile sensations. Other sources of input related to the tactile sense are those of the haptic and priopreceptive 'senses'. The haptic system deals with sensing forces applied to the body, and can inform us of approximate masses and mass distributions of objects. Currently this is difficult to simulate in a virtual environment, but some systems have been developed that provide force feedback to the user. Priopreceptive cues inform us of tensions between muscles that indicate (for example) that the arm is being moved up.

The tactile, haptic and priopreceptive senses are just beginning to enter into the virtual environment scene, with a few laboratory and commercial products becoming available. Some examples of these are the Xtensory Tactools system, which uses shape-memory alloy to stimulate fingertips [Xte92]; the Joystring, a device similar to a joystick, but with force-feedback in three dimensions [Fol87]; the Argonne Remote Manipulator, or ARM designed by engineers at Argonne National Laboratories. This is a force feedback device, providing the user with mechanical resistance to their hand and arm motions. The ARM system is used in molecular docking applications at the University of North Carolina, Chapel Hill, to provide feedback of the molecular forces involved in virtual chemical synthesis experiments (see Section 2.6); and a laboratory system for simulation of textures of surfaces (e.g. sandpaper) discussed in Minsky et al. [MOyS+90]. Minsky uses a force feedback joystick, controlled by a simulation model of the forces present on a user's finger if they rubbed their finger over sandpaper, for example (see [AB92, Rhe91, Kru91] for other information on tactile and force feedback systems).

Other senses

Other senses can be stimulated in virtual environments, to improve the immersive qualities of the system. The visual illusion of motion can be augmented by placing the user on a motion platform. This has been performed primarily in aircraft simulators to enhance the illusion of the aircraft moving, and other systems, such as the Disney Star-Tours theme-park ride (theme-parks are a common place for systems that move users around in some form). Another sense that has been stimulated in virtual environment systems to enhance the immersive effect is that of smell. Morton Heilig's Sensorama [Rhe91, Kru91], is an example of a system utilising the sense of smell. The Sensorama combines a stereoscopic film loop of a motorcycle ride through Manhattan, with motion cues of the motorcycle's movements and odour cues of pizza and exhaust fumes presented at appropriate moments. Adding feedback devices to convey temperature of objects in the virtual environment is another possible method of increasing the immersive feel of the virtual environment. Taste is one sense that has not been explored a great deal as yet.

3.3 Tracking devices

Most of the above input and output systems depend on some form of tracking device(s) to determine where a user's limb currently is in the real world, or what direction their head is facing, for example. Tracking devices are usually required to report either the position of a particular object in up to 3 dimensions, the orientation of the object, also in up to 3 dimensions, or a combination of both of these, giving up to 6 degrees of freedom — location in a 3 dimensional space in some arbitrary X, Y and Z axes, and rotation around these three axes to give *pitch*, *yaw* and *roll* of the object. A particular tracking device may report either the position of an object, the orientation of this object, or both its position and orientation; in the following description, 'position' has been used to cover all three of the above cases. For a more in-depth description of tracking devices and their behaviour, see [MAB92, App92, AB92].

Tracking devices are composed of a number of sections, depending on what form of tracking system is being used. The configuration of sections in the tracking system classifies the system into one of 2 categories, *active tracking* or *passive tracking*.

In an active tracking system, the tracker transmits a known signal, and variations in the received signal are used to compute position. This requires the following configuration of tracking devices:

$$emitter \overset{signal}{\rightsquigarrow} sensor \Rightarrow signal\ processing \Rightarrow output$$

Passive tracking systems on the other hand analyse some feature of the tracker's working volume which changes according to the tracked object's position or orientation within the working volume. For this configuration, the following devices are typically used:

$$\overset{signal}{\rightsquigarrow} sensor \Rightarrow signal\ processing \Rightarrow output$$

The tracking system components may be configured in various ways, with the sensors on the tracked object and emitters, if used, pointing into the working volume from the perimeter (known as *inside-out*), or with the emitter, if used, on the tracked object and the sensors on the perimeter viewing the working volume (*outside-in*).

For any particular tracking device, a number of attributes exist. These attributes need to be evaluated when choosing tracking devices for a particular application, as the 'ideal' tracking device is not yet invented, and current devices have many inadequacies that separate them in terms of usefulness.

Working volume

The range of three-dimensional space over which the tracking device reports position accurately is the device's working volume. This should be as large as is practically possible, to allow freedom of movement through the real world corresponding to freedom of movement in the virtual world. Some forms of tracker have large operating volumes but only under certain conditions. An example of this is acoustic trackers using ultrasonic signals to determine the tracked object's position. Acoustic tracking systems are typically active trackers, sending out an audio signal via transducers such as speakers, and determining the tracked object's position from some aspect of the signal received via microphones. Ultrasonic transducers, emitting and sensitive to frequencies in the 20kHz to 100kHz range, are normally used in acoustic tracking systems because of the small wavelength of the signal, which aids the accuracy of the tracking system, the high frequency being inaudible and not distracting to the user.

Ultrasonic transducers typically have a very small beam width making them very directional (see Figure 3.5) and although the emitter(s) and/or sensor(s) may be moved reasonably large distances apart relative to each other, they must be pointing towards the other device (i.e. emitter pointing to sensor, and sensor pointing to emitter), to work

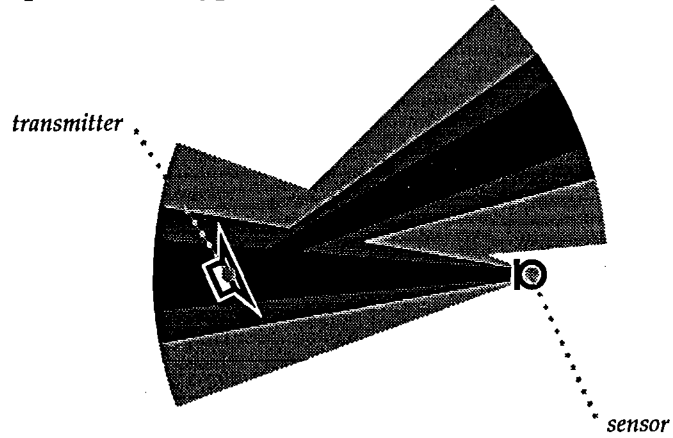


Figure 3.5: Ultrasonic transducer and sensor beam-width

effectively. Otherwise the signal is too faint to be picked up, and as a consequence, tracking will be lost.

The fitness of the tracking system to track multiple objects is related to working volume. Many virtual reality applications require that multiple objects (such as one or both of the user's hands, their head, and possibly their body as well) be tracked. Linkages to trackers can get tangled, beam paths can intersect each other causing interference, and other scenarios of a similar nature can occur.

Accuracy and resolution

Resolution is the smallest unit that can be reported from the tracker, whereas accuracy is the smallest unit that can be reported *correctly* from the tracker. A high degree of accuracy is preferable to a lower one. An accuracy rating higher than the average movement or rotation of the tracker under normal use, will probably be insufficient for the application. Consider the case where movements are in the order of millimetres, and the tracker's accuracy is in the order of centimetres. This will result in values caused by movements to be indistinguishable from values caused by noise in the system. Resolution differs from accuracy, as various effects can cause the reported position of the tracked object to drift around a 'central' value. As an example, the tracking system may be an ultrasonic based time-of-flight system (see below for a description of time-of-flight systems), with a resolution of 3 millimetres. However, due to environmental noise the reported values from the tracking system may only be accurate to $\pm$ 1 centimetre. Improving the resolution generally improves the accuracy as well; signal processing such as averaging of the positions can achieve a higher degree of 'pseudo'-accuracy.

Registration and stability

The tracking device should match real world movements accurately, in that reported relative movements and positions should match physical relative movements and positions. A movement of three metres should be reported as a change in the position of the object of three metres. If instead, the tracking device reports a change in position of 2.7 metres, then discrepancies begin to occur, and cumulative effects after a period of time cause the reported position of an object to be vastly different from the actual position. Positional drift of an object can be acceptable in some cases, when the user is totally immersed in a virtual environment, but in other cases it can be a crucial factor. An example of this is in the case of augmented reality, where a virtual environment is used in conjunction with real objects. It would not be visually satisfactory to have a virtual wall intersecting a real chair at an odd angle (see [MAB92] for more examples).

Another related topic is how much these values oscillate between adjacent values because of inability to resolve the value further. This is equivalent to quantisation noise, and is more noticeable if the tracker's resolution is poor. Stability is a major issue with trackers in use on HMDs, due to the synthesised picture information being presented to the user. If the stability of the tracker is poor, the computer will interpret variations in the data as being movements of the user's head, requiring updating of the image. If, however the user is actually not moving they will see the images before them move and 'jitter' about, depending on the aspect of the tracker that is not stable. This usually leads to feelings of nausea and headaches, as it is not a common occurrence when viewing the real world (see Section 2.2.1, on Visually Induced Motion Sickness). Jittering of other tracked objects such as the user's hand can have negative effects on the immersive experience, particularly in an application requiring fine dextrous control and and accurate placement of virtual objects.

Sample, data and update rates

The sample rate of a tracking system refers to the rate at which sensors are checked for new values. The tracker's data rate is the rate of computed positions per second, and the update rate is the rate at which position values are reported to the 'host' computing system. A higher rate is preferable to a lower one for all three of these measurements, as a high rate of values being reported (i.e. data about a position being sensed, converted and passed to the host system) gives the host more recent data than it would have if a lower rate were used. A high update rate can be sacrificed to achieve better accuracy, when the data is averaged for example.

Lag, also known as 'latency', is a measure of the delay between a change in the tracker's position, and the change being reported to the host computer. Due to the time involved in processing of the sensor's output signal and converting it to a position, the tracked object's new position may not be rendered until some time after the object has moved. If the sample rate is low, the data output from the tracking system will be updated slowly. If signal processing is applied to the data, then changes in position of the tracker will take even longer to propagate through to the host machine. From this example it can be seen that the lag of a tracking device can only be as good as its sample rate, and is typically worse. This problem is very common in electromagnetic tracking systems due to the large amount of signal processing involved with these trackers (see signal-strength tracking devices in Section 3.4).

Other considerations

How much the tracking device constrains the user through physical limitations such as mechanical linkages and cables may be important for example in some applications requiring a large amount of 'unique user usage' (many different individuals using the system in a short period of time). For applications that require a high degree of rapid movement, physical linkages can be detrimental to the general behaviour of the system. Inertial aspects of mechanical trackers, and connecting cables of other tracker types can constrict the user. The tracking device may only be movable in a fixed number of directions, or up to a maximum speed, which may affect how the user has to interact with the system.

Cost, maintenance and life-time of the tracking system needs to be considered in certain applications. Those that have a mechanical component will wear out with use, and need to be replaced or repaired. An image analysis tracking system that is in a passive, outside-in configuration and has a much lower number of moving parts is likely to last longer than a mechanical tracking system. This is especially true for high-usage applications where objects are being tracked often or continuously, for long periods of time.

When evaluating any tracking system, each of the above attributes needs to be considered. The tracking device most suitable for an application depends on the requirements of the application. For example, a desktop application where the user is seated, wearing a head mounted display, could make use of a mechanical tracking system, because this kind of application typically does not require a large working volume, but may require a high degree of accuracy and high update rate. Whether to use inside-out or outside-in, passive or active tracking can greatly alter the cost of the tracking system. This cost is also affected by the choice of tracking technology used.

3.4 Current tracking technology

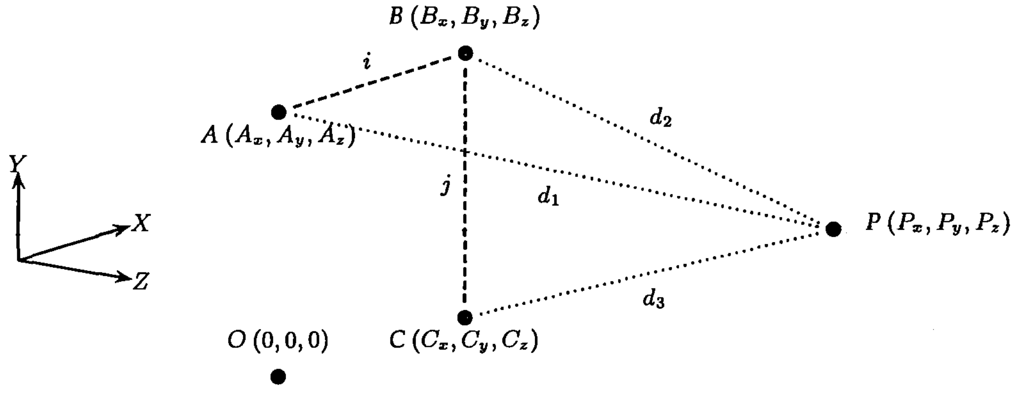
A wide variety of tracking methods are currently available, each with its own advantages and disadvantages. Most of these methods depend on some form of modulation of a known signal, to produce a measurable variation in the signal following a change in the tracked object's position.

The most common forms of tracking devices utilise various aspects of a particular medium or mediums to determine changes in the tracking device's position. Currently, the most popular technologies are optical, aural or acoustic, mechanical, and magnetic. A variety of methods are available for most of the technologies to determine an object's position [App92]:

Time-of-Flight : The time taken for a signal to travel from a transmitter to a receiver is measured. This method relies on a slow propagation time through the medium. Acoustic systems are a suitable technology for this method of tracking, as the speed of sound waves is approximately 330m/s in air, and 1500m/s in water. Since the propagation speed of the medium is known, a distance propagated can be determined from the time-of-flight measurement. To correctly determine the position of a tracked object in 3 dimensional space, four sensors must be tracking it, although three can be used if the object will never move from one side of the sensor plane to the other (see Figure 3.6).

To determine all three axes of orientation of the object, it must have 3 sensors mounted on it. From this configuration of sensors, the object can be tracked correctly. Fewer sensors either completely remove degrees of freedom, or introduce ambiguities into the reported values. These ambiguities can be ignored in certain cases (such as ignoring the possibility of the object going 'behind' the sensors, if the sensors will be mounted on the wall and it is not physically possible to move past them).

An electromagnetic system, using visible light, radio waves, or any other form of electromagnetic waves is not as suitable as acoustic signals for this tracking method, as the propagation speed is in the order of 300,000 kilometres per second, which gives rise to extremely short time-of-flight periods when measured over small distances



$$
\textit{where:}\quad
\begin{aligned}
P_x &= \frac{d_1 - d_2 + i^2}{2i}\\
P_y &= \frac{d_2 - d_3 - j^2}{-2j}\\
P_z &= \sqrt{|d_1 - P_x^2 - (P_y - j)^2|}
\end{aligned}
$$

O, A, B, C are co-planar

Figure 3.6: Determination of a point in 3 dimensions by triangulation

(a few millimetres, for example). The Nintendo PowerGlove uses the time-of-flight method with ultrasonic transmitters and receivers to determine position [Egl90]. Ultrasonic devices suitable for this application are quite cheap, and software to triangulate the signals to obtain 3 dimensional information is straightforward.

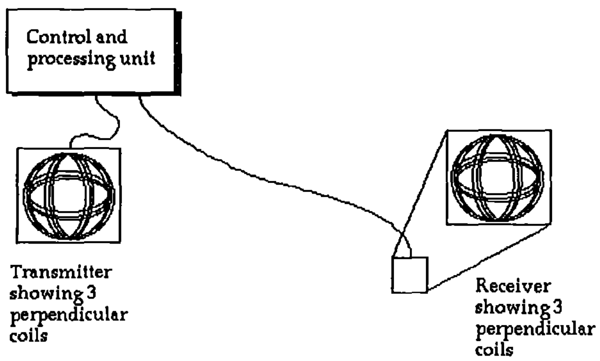


Figure 3.7: Polhemus electromagnetic tracker

Signal strength : A received signal is compared to a known signal, and the difference in signal strength used to determine angles and/or distances. The signal may be some form of light (infra-red, ultra-violet, visible), electro-magnetic radio waves, audio, or possibly magnetic field variation. The determination of a 3 dimensional position can be done in a similar way to the time-of-flight system. One notable example of a signal strength based system is the Polhemus 'Isotrack' and variants (see Figure 3.7). Due to the design of the 'Isotrack', analysis of the received signal determines both distance and orientation. The PLADAR system, based partially on signal strength and partially on direction is discussed in [Fen93].

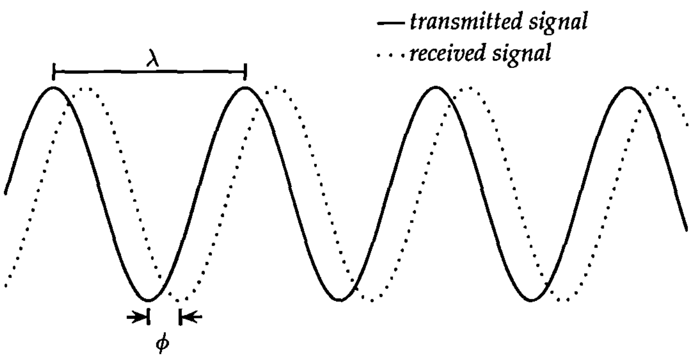
Signal strength based systems are prone to environmental interference, and the distance of the tracking device from the sensor needs to be accounted for (a signal radiating omnidirectionally diminishes in strength at a rate of $1/r^2$ where r is the length of the signal path from the transmitter to the sensor). Calibration is also required, as the sensor's environment changes and this can affect the sensor's readings. An example of this is a large ferrous object in the vicinity of an electromagnetic tracking system. Such an object distorts the magnetic field generated by the system and sensor readings vary from those that would be obtained if the ferrous object were not present.

Direction can also be used as a form of triangulation, similar to that used by the previous two systems. Sensors are available that can report the angle of incidence of a light source to the sensor's surface. A number of angles, from two or more sensors allows the determination of the object being tracked [App92, WRB$^+$90, MAB92, Fen93].

Image analysis and optical : One or more 2 dimensional cameras view the operating volume and an image processing system determines the whereabouts of various objects, using edge detection, image enhancement, and a number of other image processing algorithms based on the signals from each camera. Such systems are usually passive, and outside-in configurations have the advantage of being 'unencumbering' — they do not require the object being tracked to have any other devices attached to it. This kind of system is the basis for "Videoplace" (see Section 4.1). Image analysis is subject to the available lighting conditions of the working volume, and inadequate lighting can severely decrease the quality of the image being analysed. As a consequence tracking data can vary greatly in quality. Occlusion of the object by another object is also a major problem with image analysis systems. Other forms of optical tracker have been used to determine positions based on angle of incidence of a light source on a sensor, and using some form of camera device. For example, the system described by Fuchs et al. [FDJ77] uses a one dimensional

Charge-Coupled Device (CCD) camera as its sensing system, and the Twinkle Box device built by Burton et al. [BS74] uses an arrangement of lenses, a disk with radial slits, and a photomultiplier tube to sense a spot of light being tracked.

Mechanical : Some physical linkage to the object being tracked has sensing devices on it. A mechanical tracking system has the advantage of being fast and accurate, but is cumbersome and limits the movement of the object being tracked. Construction of a mechanical tracker is very simple, with few components, and the software to determine position is also reasonably straightforward. An early example of such a system is Ivan Sutherland's "Sword of Damocles", so named because the mechanical linkage for the tracking system hung down from the ceiling above the user's head. Other examples are mentioned in [AB92, App92, Sut68, Fol87].



where changes in ϕ determine changes in position, in accordance with $\Delta d = \frac{\Delta\phi\lambda}{2\pi}$

Figure 3.8: Phase reference tracking method

Phase-Reference : A comparison in the phase of two signals (either electromagnetic or acoustic) determines a relative distance (see Figure 3.8). A continuous signal is sent from the emitter to one or more sensors. The receiver unit then determines a relative phase difference between the emitted signal and the received signal. A phase-reference system can give very fast update rates, since the signal sent from emitter to sensor is usually continuous and changes in the position of the tracked object are instantaneously reflected in changes in the phase difference of the transmitted and received signals. The phase-reference approach gives a relative measure of change; as the emitter and sensor move away or towards each other the phase difference between the two signals changes, and as such the system is not capable of absolute distance measurements. Phase-reference systems are prone to drift over time, with the cumulative effects of errors in the positions reported. This type of system was also tried in Sutherland's head-mounted display system [Sut68].

The ideal tracking device is likely to have the following attributes [App92, MAB92]:

- a large working volume, to track users wherever they can physically move while in the virtual environment;
- portability;
- ability to operate in almost any environmental conditions (such as near large metallic objects or electrically noisy devices);
- multi-participant capability (a number of objects all being tracked by the same tracking system, without interference).

Such a tracking system is unlikely to be provided by any single technology and process , but rather a combination of two or more technologies, with each system handling weaknesses of the other system or systems.

3.5 Summary

A number of devices are available to create a virtual reality experience. Some of these devices are designed to be worn by the user for that user to experience immersion, while other devices have a passive nature and carry data to and from the user by non-obtrusive means. Of the glove devices that are worn by the user, most require one or more tracking devices to determine the position and orientation in a three dimensional area of the part of the user being tracked.

Chapter 4

Related Work

A number of novel systems for both musical performance and virtual reality have been created and used by various researchers and musical performers. Some of these are used for public exhibitions as well as research, such as the Videoplace-based systems of Myron Krueger, while others are simply one-off performance devices, that were assembled for the sole purpose of creating a novel musical instrument. Such instruments are most often created by the performer for their use, and are not in production as a consumer item, nor are their findings concerning the design and use of the instrument discussed in research journals. Many of the systems mentioned below use the Musical Instrument Digital Interface (MIDI) standard for controlling musical synthesisers, and in some cases, other devices. This standard was developed to provide a common interface between electronic musical devices to replace a number of manufacturer-specific standards in use up till that time. For an in-depth description of the MIDI standard, the reader is referred to Rumsey [Rum90].

4.1 Videoplace and friends

Myron Krueger began work in the early 1970s on various forms of human-computer interaction [Kru91, Kru83, Rhe91]. His goals were to produce unencumbered artificial realities, where people could experience and interact with a computer or, through the computer, another human in an environment that was abstract and arbitrary. He developed an image processing system to aid in this interaction that by use of video cameras and special purpose hardware could extract the image of one or more participants from a simple scene and present that image in some interesting way. Some of his initial works in this area, *PSYCHIC SPACE* and *METAPLAY*, were situated in public exhibitions, allowing visitors to interact with each other, an operator, a computer, or some combination of these. PSYCHIC SPACE included sensors on the floor as part of the interface to the computer, and stepping on various sensors would produce musical tones, but primarily the systems offered only visual feedback.

In more recent work Krueger has developed Videoplace and Videodesk, systems based on the image analysis techniques used in PSYCHIC SPACE and METAPLAY, that can run various 'applications'. Figure 3.3 in Chapter 3 shows the application "Critter", where a computer generated character interacts with a human participant. In Figure 4.1 a Videoplace application called "Mandala" is shown. Mandala produces kaleidoscopic pictures of the user on screen by displaying an image from a live video source on the screen in eight locations, each one rotated about a central axis point. A system based on the Videoplace concept and using relatively low cost off-the-shelf hardware is available commercially from the company Vivid Effects. This system allows the user to control their image on-screen to interact with drums and other percussion instruments. When the image of the user's image touches a drum image onscreen, MIDI commands are sent to a synthesiser to produce sounds (see Figure 4.2).

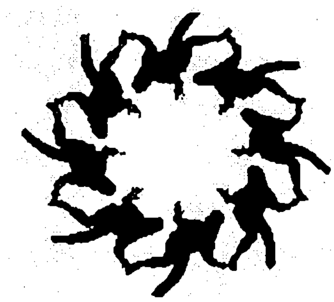


Figure 4.1: Videoplace Mandala application

Kreuger's work on artificial reality is non-inclusive; his aim of creating un-encumbering artificial realities has caused him to create "Responsive environments", where the surroundings of the participant change and adapt to the participant's movement or lack of movement. This contrasts with what Jaron Lanier, former head of VPL (manufacturers of the DataGlove and DataSuit), defines 'virtual reality' to be, where the user by wearing goggles, gloves and other devices is 'immersed' into an artificial environment.

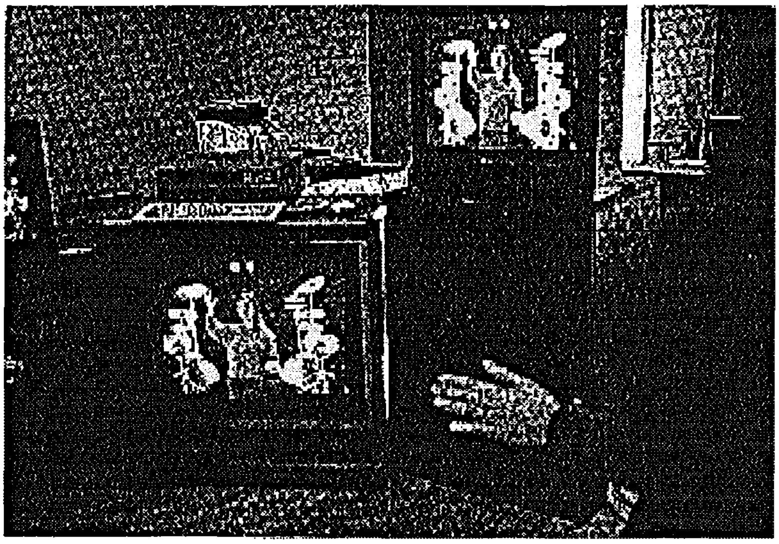


Figure 4.2: The Mandala system, based on *Videoplace*

4.2 Theremin

The Theremin is an instrument designed by Leon Theremin in the early 1900s, consisting of a box with two aerials extending from it, that each radiate a low power high frequency field [Rhe]. One aerial controls the pitch of played notes, and the other controls the amplitude of the notes. A notable performer of this instrument was Clara Rockmore who is shown with a Theremin in Figure 4.3. A virtual "Theremin" was implemented by Mark Bolas and Phil Stone while at NASA Ames [PT92]. In their system the user can modify the pitch of a note and change the character of the sound by using a glove.

4.3 Gesture And Media System

The Gesture And Media System, or GAMS, is a wand equipped with ultrasonic sensors that a performer holds while moving about in a sensing area [BF92]. It uses a 68HC11 microcontroller to time the period of travel of pulses from four high frequency speakers mounted above the performance area to the wand held by the performer. These distances are fed to a computer where mapping software translates positions of the wand into various MIDI events and outputs these to a MIDI device for the performance. Various MIDI events can be controlled by the system, and an editor is used to determine how regions of space will map onto MIDI commands.

Figure 4.3: The Theremin

4.4 Modular feedback keyboard

The Modular Feedback Keyboard, developed at ACROE, is a device that simulates the tactile and haptic feedback of a large number of instruments [CLF90]. Figure 4.4 shows the modular feedback keyboard in a piano/organ-style configuration, although other configurations are possible. The device simulates the forces and sensations that are present when playing an instrument.

This device is based on an earlier system known as the *Cordis* system, a device which simulates various instruments by responding to the gestures of the performer in the way a particular real-life instrument would respond. Cadoz et al. [CLF84] describe analysis of the mechanical vibration and excitation structures of musical instruments, and their behaviour when the instrument is being played by a performer.

4.5 Gustav's Party

A rock band known as "Gustav's Party" have recorded a music album of a virtual reality experience, where objects in the virtual environment can "be made to sound like a drum when hit" [New92]. The band uses three dimensional audio technology to give sounds in the recording of apparent positions in a volume, instead of the planar effect that is achieved with conventional stereophonic recording. According to the band their recording will "probably require special devices to be heard and seen, perhaps in video arcades".

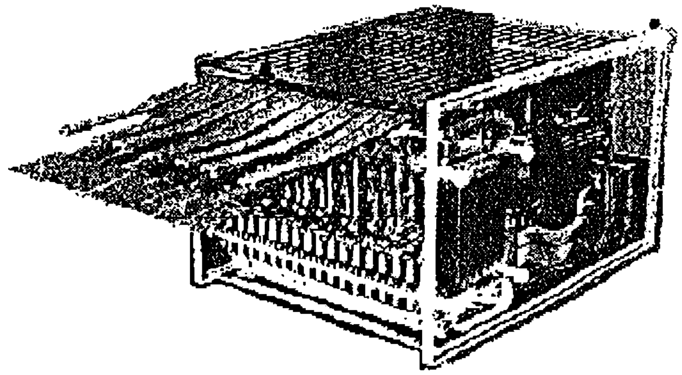


Figure 4.4: Modular Feedback Keyboard

4.6 The VideoHarp

The VideoHarp built by Dean Rubine and Paul McAvinney [RM90], is a harp-like instrument that determines where the musician's fingers are on a two dimensional surface, and translates the position to MIDI commands. The VideoHarp is one example of what Rubine et al. term a *programmable finger-tracking controller.* Such a device was conceived by Rubine et al. to allow performers to experiment with new playing techniques, rather than use instrument controllers that mimic existing instruments and their methods of play. The reasoning behind this is that instrument controllers based on existing instruments and triggered in the same way to those instruments, are likely to be used by the performer in the same way as the performer would use the original instrument, with similar styles of music produced. Basing a new instrument controller on the motions and other related features of a performer's fingers, the programmable finger-tracking device can be "programmed to respond to many different types of gesture; in effect such an instrument could act like many different instruments" [RM90].

4.7 Radio Baton and Conductor program

These two related systems were developed to tolerate a low level of skill on the part of the performer, to operate electronic synthesisers in real-time performances [Mat91]. The aim of the Conductor program, and the Radio Baton used to control it, is to allow the performer to concentrate on the expressive parts of a musical piece rather than the 'lower-level' technical aspect of operating the instruments. The Radio Baton is a device in the form of a baton, that contains low-frequency radio transmitters in its ends, and is sensed by an array of receiving antennae. The Conductor program receives position signals from Radio Batons and translates them to MIDI command signals to be fed to a synthesiser. This program functions as a sequencer, with timing controlled by input movements from the baton.

4.8 PreFORM

A gesture based performance system is discussed by Chabot [Cha90], that includes the notion of 'virtual machines' in the design of a computer based music system. The system discussed in the article uses these virtual machines to aid in creating a device independent music system that takes inputs from a number of different devices, and maps the inputs to MIDI commands. The virtual machine concept allows a generic interface between input and output devices and the mapping system itself.

Currently the system makes use of a SONAR device, which maps distance to a MIDI parameter, 'air drums' (sticks containing accelerometers, which report sharp linear and angular accelerations; the acceleration values are mapped), a pendulum device that maps angular values to MIDI parameters, and a pitch tracking sub-system, where information on pitch, attack rate, amplitude and inflection of an input sound (in this case from a flute), is used for mapping. A discussion is given as to why gesture based performance systems should utilise a physical volume as the performance space; Chabot does not consider switches and buttons 'gesture interfaces' due to their lack of defining a 'space' for the performance.

4.9 BioMuse

BioMuse, developed by Knapp and Lusted [KL90, Gib93], is a spinoff from biofeedback technology that uses electrodes placed on various body surfaces to detect brain waves, eye and limb movements. The musical application of this technology is a system that maps signals from the EEG (electroencephalogram, or brain waves), EMG (electromyogram, or muscular activity), and EOG (electrooculogram, eye movements) sensors, to various MIDI parameters

An example of the system in use is one where a 'virtual' violin can be played, with one set of EMG signals controlling the pitch and vibrato of a playing note, and another set of EMG signals from the other arm controlling *Note-On* commands and the amplitude of the played note. The performer can 'steer' the sound around the auditorium by use of the EOG signals; for example, looking to the left pans the sound to the left.

4.10 Magnetic gloves

The performer Laetitia Sonami has used gloves equipped with magnetic field sensors to control MIDI instruments [Gar92]. The gloves are fitted with sensors that detect the magnetic field between her thumb and fingers and translate the measurements into MIDI commands.

The nature of the performance is not specified in [Gar92] (i.e. no mention of simulating existing instruments is given), nor does it give details of what MIDI commands are sent by the glove devices (these commands may just be simple NOTE ON / NOTE OFF commands, or they could include PITCH WHEEL commands for example). The review does not suggest that any form of position tracking is employed. Tracking of some sort, or reporting of relative movements of the hands, would be needed to simulate many instruments. The MAX development software from Opcode Systems (see Chapter 6 and [CMJ91] for a short description) is used to create the basic musical structure of a piece. Parameters of the piece are then varied during the performance, by use of the gloves as well as the performer's voice.

4.11 MusicWorld

MusicWorld is an application developed as an example of a system that makes use of predictive Kalman filtering [FSP92]. Friedmann et al. [FSP92] are researching the effects of synchronisation of events in virtual worlds. Such events could be, as in the case of the MusicWorld system, the visible display of a virtual drumstick, under the control of a user, striking a virtual drum, and the triggering of a percussion sound from a synthesiser that the user hears. Because of large lag times with many tracking systems (in the order of 20msec or more in come cases), and delays in rendering the altered scene, the user's actions can lead the on-screen, audio and other display system responses by a significant and noticeable amount, disrupting the user's experience of the virtual world.

4.12 Bug-Mudra

Tod Machover at the MIT Media Lab has developed a system known as *Bug-Mudra*, that uses a glove device for live alteration of performances by human players [Stu91]. In

Bug-Mudra, an Exos Dextrous Hand Master (DHM; see Figure 3.1) is used to vary parameters on a MIDI mixing board, into which audio signals are fed from MIDI synthesisers. Instruments played by the performers during the performance send MIDI commands to the synthesisers. A conductor directs the performers, as well as controlling the MIDI mixer parameters via the DHM. The parameters that can be changed include amplitudes, tonal qualities, and left-right panning of input signals.

4.13 VR exhibits

A virtual reality system installed at the Computer Museum in Boston allows participants to create art or architecture through the use of some input devices and view them interactively on an HMD [PT92]. Participants can create three dimensional drawings in the air, with sound effects added as well. By waving a sensor-equipped wand, the user creates a trail of glowing objects, and as they are created musical tones are generated by the system, and localised in the approximate position where the objects are being created. Waving the wand around your head causes the sounds to move around your head as well. Users of the system can also assemble a house from predetermined blocks consisting of walls, windows and pillars, in another application. "The wand grabs building blocks with a touch and moves them wherever you want".

At the California Museum of Science and Industry, a virtual volleyball system is available where participants control onscreen players in a game of volleyball [Poo94]. This system is very similar to Myron Krueger's Videoplace, and Vivid Effects' Mandala, both mentioned in Section 4.1. Other displays in the exhibit include examples of image morphing; an HMD and sensor system that creates the "illusion of flying through the air".

4.14 Others

A comprehensive survey of early computer–music interfaces is presented in [Pen85]. This article discusses many GUI based systems mostly for offline or noninteractive performance. There is some mention of systems just coming to the marketplace (1985) that allow real-time alteration of musical parameters, or are computerised synthesisers of traditional analogue instruments.

Chapter 5

A Low Cost Virtual Reality Exhibit

Virtual Reality is a field in its infancy, for although the concepts have been present since the 1960s, the technology capable of achieving it to any degree has only recently become available. Currently there are only a handful of research institutes and commercial organisations experimenting in the field. One of the major reasons for the small number of researchers and users of VR systems is the high price of the equipment involved. A conflict exists in the form of price versus performance, a common problem with new technology based industries. Current computing power is insufficient to generate high quality displays readily, so expensive workstations are often required.

Input and output devices have had years of research put into them, and the manufacturers are wishing to recover the high research and development expenditure, so such devices are very expensive. Added to that is the fact that most of these devices are intended for a laboratory environment, and so are highly accurate but very fragile, which makes them generally unsuitable for mass-user environments. To have a number of gloves wear out and become unusable, at the cost of a few thousand dollars each, in the space of one or two weeks, makes the use of such gloves prohibitively expensive for such a place as a publicly funded science centre.

When designing an exhibit for an educational science centre environment, a number of factors need to be considered. Some of these factors are related to how the exhibit functions in serving its purpose of educating visitors to the centre and the other factors are concerned with the design itself.

The first group of factors is related to the exhibit's concept. From a science centre's perspective, how the exhibit's concept fits into a science centre is of primary concern. The attractive power of the exhibit — how well it draws interest from science centre visitors — is important. There should be as little bias as possible in the group of visitors that use the exhibit. Ideally, no social group or groups would be favoured or disadvantaged over other social groups. The background of a visitor should have as little impact on their use of the exhibit as possible, and they should not be disadvantaged because of their race, age, level of education or other classifying attributes.

We also need to consider how long will a typical visitor use the exhibit before they become bored or satisfied with it and move on. The 'period of interest' of the exhibit — the length of time a user's interest is maintained while interacting with the exhibit, also known as the holding power of the exhibit — should be flexible in length (i.e. during a busy period at the science centre, the user should only use the exhibit for a short time; in less busy periods the user should be able to use the exhibit for a longer time, without losing interest). The exhibit should do something interesting as soon as a visitor starts using it, otherwise the visitor may very quickly lose interest with the response that "It doesn't do anything!".

In its functioning, the exhibit should be as autonomous as possible. To require constant intervention or aid from exhibit demonstrators draws these people away from other tasks, and it is not efficient to require one demonstrator per exhibit. The exhibit should be able to function almost entirely on its own, including handling problems that the user may have caused, or that may have been caused by some other factor such as power restoration that occurs when the exhibit is switched on. Exhibits requiring power are usually connected to a central switching board; all of them are powered up and down together, and the exhibit should not require individual attention every time it is to be switched on or off.

The educational effect on visitors is important; the exhibit should have a positive educational effect on its users. A negative effect will make them less inclined to use it, and create negative publicity for either the exhibit, the science centre, or both. The exhibit should be unambiguous about what it is displaying or representing. If the exhibit teaches users the right concept poorly, or the wrong concept, then the exhibit has been badly designed. This situation may occur in a number of cases, such as incorrect instructions, or inappropriate functioning of the exhibit.

The second group of factors that is the concern of the exhibit's designer covers the implementation of the exhibit. To aid development and future enhancement or maintenance, the system should be highly modular in design. To this end, the software involved should be as device independent as possible, with little reliance on machine-specific features to improve performance, for example. The hardware should use a common interface standard between each component. An example of this is the choice of RS-232 serial communication rather than the more machine-specific Appletalk system.

Costs should be kept low, as some science centres are operated as trusts and receive the bulk of their funding from donations and grants. The cost of an exhibit includes not only the initial cost of its construction, but costs incurred over time due to maintenance and the amortised costs of employing demonstrators to explain the exhibits on the exhibit floor.

Maintenance and robustness of the exhibit needs to be considered very carefully. Exhibits that require little or no maintenance are of more value in the long term than those that require constant maintenance, because the cost of maintaining the exhibit can grow very quickly. About 70 exhibits are demonstrated on the exhibit floor of *Science Alive!*, so the cost of maintenance needs to be considered. The replacement value of components should also be evaluated. An expensive but readily available item may in some cases be a more appropriate choice than an inexpensive but rare version of the item.

In creating an exhibit each of the above factors should be evaluated at the appropriate stage in the exhibit's development.

5.1 Designing an Exhibit

In designing an exhibit for a science centre, as in designing almost any system, there are a number of stages in the process. Very rarely does a system function as desired if it has simply been conceived, implemented, and put into use. This is because of a number of possible reasons, any or all of which may be present. Unforeseen problems, underestimation of environmental conditions, and lack of planning can all cause an exhibit with a very good concept to function poorly. It is very rare for the exhibit designer to cover every possible situation in the design of their exhibit. There could be one or maybe several minds working on the development of the exhibit, but the users in the general public make up thousands of minds and the chances are very high that someone using the exhibit will conceive a way of mis-using the exhibit that the designer had not. Often a number of testing and redesign iterations are required to ensure that the system performs its desired functions. The testing stages should ideally reproduce the final operating conditions as accurately as possible to ensure the test's validity.

In the design of a science centre exhibit the following phases can be identified:

1. Initial concept - the basic theme and design of the exhibit are conceived;

2. Proof of concept prototype - build a simple or scaled-down but functional laboratory version to test the concept and gain a rough understanding of problems and costs involved in a final version;

3. "Exhibitising" - putting in all the features that are required for an autonomous stand-alone exhibit;
4. Trial runs - with "test visitors" under exhibit floor conditions;
5. Redesigning - if necessary (steps 4 and 5 are repeated as many times as are feasible or required to obtain satisfactory performance for the invested cost);
6. Placement on exhibit floor;
7. Maintenance and updating of exhibit.

Some of these phases will be covered in greater detail below, with reference to the design of the "Virtual Maestro" exhibit for *Science Alive!* as an example.

5.2 Initial concept

A virtual reality system was chosen as the theme of the exhibit, its function being to educate visitors to the science centre *Science Alive!* about virtual reality. Most visitors to the science centre had never used virtual reality systems before, and although some visitors had knowledge of virtual reality and its capabilities, this knowledge varied from small articles in the popular press to demonstrations shown in science oriented television programmes. For the science centre exhibit, visitors would learn about VR firsthand, find out what it is, how it is achieved and what can be done with it. The basic function of a VR system is to simulate an aspect of reality in some way. Musical instruments were chosen as being the aspect of reality to be simulated, where virtual instruments are created in a three dimensional space in front of the user who can play them via a glove device connected to a computer. The glove device reports information to the host computer on the position, orientation and finger status of the user's hand. This information is analysed and used to generate MIDI commands (see Chapter 4 for a brief description of the MIDI standard) for a synthesiser to produce sounds.

A musical system was chosen because this allows the cost and complexity of a graphical display system to be avoided (there is no restriction on adding one but it is not crucial to the exhibit's functioning), and music-creating devices are generally popular with a wide range of social groups, irrespective of age, gender, race and level of education, although those with no formal musical training may feel disadvantaged by such a system. As an aid to this group, the instruments are designed as "novice" versions of their real-life counterparts producing "good" sounds when played, no matter how the instrument is played. Each instrument uses a C-major scale (no black notes can be played) and has very large note regions (the three dimensional region that produces a note of a particular pitch; for a piano instrument this is the size of one virtual key), to make it easier to play.

In simulating the reality of playing musical instruments the system should behave in a manner resembling the instrument it is intended to simulate, because the degree of behavioural similarity between the simulation of a object and the object itself directly affects the user's sensation of interacting with a real object, or the realism of the system.

A very simplistic system might produce sounds when the user moves their hand into, out of, or through a particular three dimensional space. The sound produced might be a tone of fixed amplitude and pitch, capturing none of the velocity, magnitude or position information of the user's hand motion. Such a system is unlikely to give the user a sensation of playing a musical instrument, but more likely to produce a feeling of "If I put my hand here, I hear a beep".

However, by adding various aspects of a piano's behaviour to the system, the required sensation is more likely to be generated. For example, modern music synthesisers can produce very realistic piano sounds. Replacing the monotonic note in the simple system mentioned above, with a piano sound produced by a synthesiser, and driving the synthesiser with signals that vary the pitch and amplitude of the sound depending on the velocity, orientation, position and finger status of the user's hand, produces a greatly enhanced sensation of playing a real piano. The goal of the project is to ensure that the user perceives to as much degree as possible, that they are producing sounds from a piano, drum kit, or some other real instrument.

The structure of the exhibit differs from common VR systems. In typical virtual reality systems a user is immersed into a virtual world via the use of a helmet (or HMD), gloves and other devices (see Chapter 2). Using a head mounted display has a number of drawbacks. Firstly helmets are expensive, in the order of US$5000 and upwards. It is unlikely that this price will drop in the near future, due to high manufacturing costs of the components and assembly of the helmet. The resolution of current liquid-crystal display based HMDs is poor, (one system in commercial production is 360 $\times$ 240 pixels [Vir92]) and cathode-ray-tube based HMDs are unacceptably heavy for small children, becoming uncomfortable to wear in a few minutes of use. Their weight is also linked to their inertia, and many users of HMDs find it unnatural to have a large mass on their head, especially when they turn their head to look in another direction. The majority of HMDs available are designed for laboratory use, and very fragile, so they would not last very long under exhibit conditions.

Other types of graphical display are possible such as large screen projections (used by Ralston [Ral94]), and normal video monitors, but any type of graphical display puts a great demand on computing power needed to display a reasonably realistic image. Pausch [Pau91] observed that representational behaviour and low lag were more important than high resolution and photorealistic appearance, for improving the realism of the system felt by the user. This has also been observed by Heeter [Hee92].

Some effects of HMDs and other displays on the user make the use of such displays undesirable. In Section 2.2.1, an introduction to simulator sickness (SS) and visually induced motion sickness (VIMS) was presented. The effects of SS and VIMS are generally unpleasant and some could be potentially life-threatening depending on the user's actions after experiencing the simulation. Some examples of these effects are eyestrain, nausea, disorientation, and fatigue, all of which may place a user at risk after use of a VR system [HR92, LKL+92, Ebe92, MS92, DL92, Bio92, PCC92]. For example, effects such as disorientation and dizziness would be extremely detrimental if the user attempts to drive a vehicle after experiencing the simulation.

The educational value of the exhibit could be affected as SS could interfere with the user's ability to learn; if they are preoccupied with SS symptoms their learning capacity will be reduced [LKL+92]. Users may reduce their 'range of use' of the VR system to avoid SS effects. Pilots in flight simulators have been known to introduce strategies for reducing SS symptoms, such as flying by instruments and restricting head movements [LKL+92, PCC92]. Also, users may adapt to the simulated VR world, but it could (and usually does) differ from the real world [LKL+92]. As a consequence, the response given by the user in the VR might not be appropriate for the real world version of the VR simulation. Aftereffects of SS and VIMS would be negative in a science centre environment (and indeed many other environments). Pausch (ibid) cites a number of reports of VIMS aftereffects lasting at least an hour after exposure to the simulation and in some cases symptoms have persisted for a number of days after exposure [PCC92]. For example, having a headache for an hour after using an exhibit will not have a positive effect on the exhibit, and driving through rush-hour traffic feeling dizzy and dis-oriented will most certainly not be positive. All of the effects of SS and VIMS will have negative effects on the popularity of both the exhibit and the science centre itself.

Hygiene is a major concern with the exhibit, as over 100,000 visitors to the science centre are expected per year. All exhibits in the centre are of a 'hands-on' interactive nature, so there is a high chance of spreading contagious diseases via contact with the exhibits. Including an HMD in the system compounds the issue of hygiene, with various pests and diseases transmitted in the hair. Simply the state of a visitor's hair itself may be sufficient to deter other visitors from using the system. Aukstakalnis (ibid) cite this example of an unpleasant experience they had regarding hygiene [AB92, pages 272-273]:

> However, virtual reality equipment is generally designed to tightly enclose our senses: eyes, ears, hands. This issue became crystal clear to us while standing in line at a computer trade show, waiting for a demonstration of a new virtual reality system. As we approached the front of the line, we noticed two people slightly ahead of us. The first was a stately looking gentleman in a fine suit. The second was a young man wearing blue jeans that covered only the essentials, what appeared to be a tie-dyed T-shirt (a closer look revealed the pattern to be stains of some sort), and hair with enough grease to repack the bearings of an automobile.
>
> As the first man prepared to put on the helmet, he pulled several long strands of hair from the helmet, noticing that other hairs had gotten lodged in the crevices and would probably have to be cut out later. After that demonstration, we watched as the greasy young man pulled the head-mounted display on. Everyone in line close enough to see him clearly had the same thought that we did. Who knew what his head would leave in the helmet, only to be rubbed into ours. Ten of us simply walked away.

Such an experience would have an adverse effect on the science centre, and the rate of it occurring would be very high. For the above stated reasons, no graphical system is used in the "Virtual Maestro" exhibit. Instead, a system was chosen that augments reality with virtual objects. The main advantages of this system over one utilising an HMD are that this system is much cheaper, there are fewer problems to be solved in the computing aspect (such as hidden line removal in the graphical display) and there is a certain amount of intrigue added to the exhibit, with only an input device visible. The user is required to 'explore' the exhibit to determine its function, rather than have everything displayed explicitly before them.

In Section 2.5, Heeter's three suggestions for a measure of presence were described. A display-less system fits into these suggestions with a strong feeling of presence in the real-world environment, because much of the reactions to the user that are considered in the social presence and environmental presence categories are provided by real-world entities, such as other visitors in the centre.

5.3 Structure of the system

The final structure of the "Virtual Maestro" exhibit is shown in Figure 5.1. The glove input device and the translation software are described in detail in Chapters 7 and 6. The output section is described below.

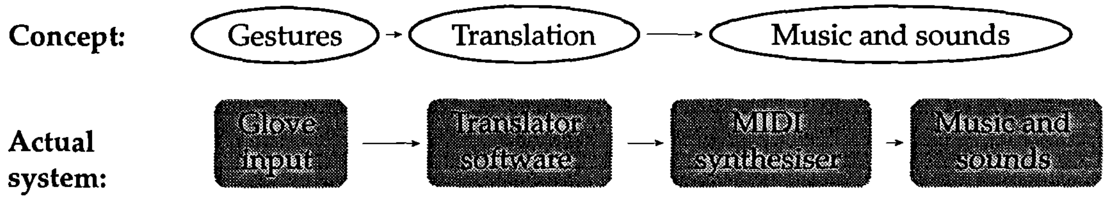


Figure 5.1: Structure of "Virtual Maestro" exhibit

Movements of the glove and its fingers by the user are translated into MIDI commands and fed to a synthesiser to produce the appropriate sounds. MIDI output is used because the MIDI specification is simple and flexible, and almost all commercially available synthesisers use this standard for communication, making it an easy operation to switch from one synthesiser to another if there is reason to do so.

The MIDI standard allows for 16 independent channels of 128 notes each, which can in theory, all be played simultaneously (limitations in the synthesiser usually prevent this however). A note is played on a MIDI synthesiser by sending a *Note On* command for a particular note and channel. It is usually terminated by sending a *Note Off* command for the same note and channel. Different instruments can be assigned to each channel, and the instrument for a given channel can be changed on-the-fly, so the synthesiser can effectively produce any of the instruments it is capable of playing at any time.

Patches on the synthesiser allow simple configuration of a number of settings at once. A patch is a setting on the synthesiser for various instrument and channel parameters that can be activated with a single MIDI command. Not all synthesisers have this feature, or even multiple instruments, and those that do not must have each channel set up individually via other MIDI commands.

The instruments modelled on the "Virtual Maestro" exhibit include a piano, saxophone, drum kit, 'string section' and 'choir'. Many of these are multi-input instruments; that is, sounds are produced by one input controlling the pitch and/or amplitude, of the note, and another input triggering the note to play and controlling its duration. An example of this kind of instrument is the saxophone, where the player controls amplitude and triggering of the note by blowing into the instrument and the pitch is controlled by fingers depressing the saxophone's keys. Pianos are a counter-example of multi-input instruments. Notes are produced by a finger striking a key; this action controls both the pitch of the note (determined by the key pressed) and the note's amplitude envelope (controlled by how hard the key is struck). The action of striking a key triggers a note to sound. The "Virtual Maestro" exhibit currently uses only one glove input device (see Chapter 7 for details) so some method is needed to use attributes of the glove's postural and gestural information to supply the other inputs required for multi-input instruments. To overcome the problem of supplying inputs not normally controlled by the right hand (the single glove for the system is right-handed), these other required inputs are mapped to unused degrees of freedom of the glove device, thus producing new instruments with similar behaviour to that of the instruments they were based upon. More details of this mapping scheme can be found in Chapter 6.

In most instruments there are what Cadoz et al. term *excitation gestures* and *modulation gestures*, for controlling the production of notes [CLF84]. Excitation gestures provide energy to the note generation device(s), causing sound to be produced, whereas modulation gestures expend little energy, and merely serve to direct or modulate the sound produced by the excitation gesture. In all traditional instruments (except for those that have a note generation system such as the organ), the performer is the source of energy for note generation, and the excitation gestures they perform are the act of drawing a bow across a violin string or pressing a key on a piano keyboard. In the "Virtual Maestro" exhibit, there is an independent source of sound generation, in the form of a musical synthesiser, that produces notes so energy is not required from excitation gestures. However, even though excitation gestures do not have to transfer energy from the user to the system, the gestures themselves have become an integral part of the playing of a particular instrument, and so they are still used in most of the instruments. The 'choir' instrument, presented in Chapter 6, is an example of an 'instrument' that uses purely modulation gestures, and requires little energy to control on the part of the user.

5.4 Other design phases

Once the concept of the system was decided upon, a simple prototype was constructed. This mostly involved development of the software for the host machine, since almost all of the hardware decisions were finalised when the basic functioning exhibit was ready. A prototype of the exhibit was built with a few instruments implemented to demonstrate the practical feasibility of the exhibit and determine what would be necessary in the final system. The prototype and concept idea were displayed to representatives from the science centre *Science Alive!* where the final exhibit was to be stationed. The prototype was a necessary development to demonstrate what is possible on a low budget. In the restricted funds situation of many science centres, it was considered a better decision financially to demonstrate a working proof-of-concept system rather than promise a possibly in achievable system and then not be able to produce it. As a result of the demonstration the science centre representatives approved the concept of the exhibit and further development and refinement was undertaken. A testing stage of the exhibit's concept was planned for local schools to gauge the exhibit's potential attractiveness to school children, the primary expected audience of the science centre. However this plan was cancelled at the science centre's request because they wished to retain the surprise element for the exhibit, and release it to the public themselves.

Initially abstract instruments were used that were similar to real-life instruments in aural properties but had a different shape, and required a different method to play them. These abstract instruments were presented to 'test visitors' before the exhibit's construction was completed and did not receive a favourable response because visitors found the instruments difficult to comprehend, and difficult to play as a consequence. The exhibit was redesigned to model real instruments more closely, and the new system received a more favourable response; this version of the system was retained.

When the exhibit was functioning satisfactorily, extra features were added to it to allow it to function autonomously. These features included automatic activation of the exhibit when switched on (some exhibits do not normally start in their 'exhibit' mode, and must be configured to activate the 'exhibit' mode upon starting up); automatic adjustment to different users (this may be required if the users have varying physical features such as hand sizes and users' heights, that the exhibit depends upon); and automatic resetting of the exhibit after a period of inactivity. The exhibit was then packaged in a case as many of the components were not to be made accessible to users. When the exhibit was ready for display it was placed on the exhibit floor, and maintained and updated as required.

Chapter 6

The Glove-data to MIDI Translator

In this chapter the main component of the "Virtual Maestro" exhibit is presented. This consists of a software translation system running on a host computer, that receives input data from an input device (typically a glove) and produces output data (currently MIDI control information) based on the input data. The software provides a mapping between input and output data which determines the behaviour of the system. A model of how an instrument is operated, and the resulting behaviour of the instrument, is used as a basis for the mapping software.

Some models are more realistic than others and a particular model of a group of gestures, positions, and possibly other input data, may poorly represent the real life instrument that the model is intended to simulate. As a consequence the model's behaviour can drastically affect the perceived realism of the instrument being played. For example, consider a piano that is modelled in the system by the following conditions:

```
play a note if below height <H>,
otherwise do nothing.
```

If the note played has a fixed pitch and amplitude, (i.e. moving below height H produces a note of the same pitch and amplitude irrelevant of position) then the system as a whole will be perceived as being more of a break-the-beam type of system, similar to door entry alarms in small shopping stores. The sensation that the user gets is one of: "If I put my hand here, I hear a sound", rather than a sensation of playing a piano. So the model used for a particular instrument should be chosen carefully as it will affect the behaviour and effectiveness of the exhibit. Modelling objects differently from their real-life counterparts has been seen in other areas of virtual reality research to be detrimental or beneficial, depending on the application. Pimentel and Teixeira report some of the findings of the U.S. Navy in their research into flight simulators [PT92]:

> The Navy discovered that the complexity of a scene (determined by the number of polygons) and texturing (each polygon is "painted" with an image) was crucial in providing adequate velocity and altitude cues for line ups and landings on both airfields and aircraft carriers. The Advanced Development Model also showed that simulation of haze significantly contributed to the realism of the experience. Generally, the more realistic, the better the training.
>
> Just to prove that there's an exception to any rule, they also noticed that pilots were having difficulties orienting themselves with respect to the featureless surface of the ocean. Their solution was to draw a checkerboard pattern on it to provide additional perspective cues–a good example of how decreasing realism can sometimes improve training effectiveness.

In another example Pimentel et al. [PT92] describe the enjoyability of virtual worlds that are based on reality, and how more realistic scenes can be less enjoyable for the user. This discovery was made by Mark Bolas, President of the VR company *Fake Space Labs*:

> At that time, even with sophisticated workstations, all [Mark Bolas] could animate was simple wireframe outlines of shapes. So he designed a suite of offices from real blueprints. At first he constructed the world without colouring in the polygons so the walls were only outlined and transparent. You could get a sense of size but also see how each room affected the other rooms if you started moving walls around.
>
> When he coloured in the walls to make the rooms more realistic all he could see was the room he was in–and viewers enjoyed the experience less! This started him wondering about the usefulness of abstraction versus realism. The coloured wall might be more realistic, but the see-through boundary lines were more useful and enjoyable for certain tasks. And by leaving the world more abstract, versus building in more realism, Mark could draw users deeper in.

As can be seen from these two examples, the level of realism needs to be chosen carefully and a level that users acceptable may differ from what the exhibit designer expects.

In designing the translator software, a number of issues need to be considered. Some of these are related to the input data and preprocessing of it before use, while others concern the functioning of the system as a whole. Some of these problems are discussed below, and solutions to some of them are presented afterwards. A number of methods for modelling virtual instruments were tried in the process of developing the system, each with drawbacks and benefits, and one was chosen for use in the "Virtual Maestro" exhibit. Features of each method are presented, to demonstrate why that method was or was not suitable for the exhibit. The two main emphases in the design of the software used are that it should represent the modelled real-life instruments reasonably accurately in their behaviour and operation, and that the code itself executes quickly and is simple in design, since the host machine may for example, be incapable of complex mathematical functions within a required time, as would probably be the case if the system were running on a small microcontroller.

The current version of the exhibit is operating on a 25MHz 486sx PC, although it has been successfully operated on lower speed machines. Use of a small inexpensive microcontroller based system would greatly reduce the cost involved, but would also reduce the flexibility of the system in some ways. An example of this inflexibility is the use of an EPROM to store the translator software for the microcontroller, rather than having the program stored on a magnetic disk, as would be the case for a larger general purpose computer. Storage on the EPROM is not as easy to change as that on a disk. Code for the microcontroller based system would need to be fast, and probably coded in assembler, both of which can reduce the flexibility of the system even further. However, once the system is finalised, the translator code may be ported to a microcontroller system to reduce the exhibit's overall cost and free the PC for other tasks.

In an effort to keep the system machine independent, it was decided that a program should be written from scratch to perform all of the input, translation and output handling of the glove and MIDI commands, rather than develop the system on a specialised toolkit. One such toolkit, available for the Apple Macintosh computer and known as "Max" by Opcode, can be used to develop MIDI applications for the Macintosh, with a high degree of flexibility in the functionality possible. However it is only available for the Macintosh and the choice of machine used in the final exhibit may change at a later date. Some other reasons for not using such toolkits are the cost involved if the toolkit must be purchased, and low level control of the computer or MIDI system may not be possible due to intervention on the part of the toolkit.

6.1 Human factors

The science centre is aiming at a broad range of visitors, with varying physical features, who will use the exhibits on display. Such features include the height and reach of the visitor and the size of their hand. As many visitors will be users of the "Virtual Maestro" exhibit the choice of size and position of the instruments will need to take user hand sizes and reaches into account. The exhibit will be operated constantly and users should be able to operate it correctly, being able to play all of the virtual instruments independent of their height and reach. In other words, the virtual instruments should be designed and placed such that all users are able to play them correctly.

Allowances in the system need to be included to cope with the variations in sizes of users' hands when operating the glove device. Small children are less likely than adults to be able to fully operate the glove device, so the system must be able to adapt to the varying size of hands in circumstances that depend on hand features, such as determining whether a finger is flexed or not.

Because of the novelty of the "Virtual Maestro" system it is unlikely that visitors to the centre will have experienced such a system before. Therefore the usage of the system should be simple and quick to learn, because potential users will lose interest quickly if they appear to be having no effect on the exhibit. Also the system should not have a negative effect on the user or the surrounding visitors so the virtual instruments may need to be 'adjusted' so as not to behave quite like their real world counterparts. An example of this is the use of continuous pitch sources for the instruments, which can generate pitches that sound unpleasant and should be avoided.

As there is no graphical feedback in the system (see Section 5.2) a simple method for the user to find the virtual instruments should be included in the system. The user will become disinterested and abandon the exhibit if they cannot get it to perform its function due to not being able to locate the virtual instruments.

Some instruments are more popular than others, such as piano, guitar, and saxophone, and some are easier to play, or to model. The representation of instruments and how their behaviour is modelled is a major design issue. As mentioned previously, modelling a real instrument closely may be undesirable or even infeasible. The exhibit currently has only one glove for input so instruments requiring two hands need to be modelled differently from their real life counterparts if they are to be included in the system. Most visitors know the approximate gestures for triggering a wide variety of real-life instruments, but few will know the correct fingering required for, say, the saxophone. Since some of the instruments in the "Virtual Maestro" exhibit will have similar behaviour and triggering gestures to their real-life counterparts, prior knowledge of how to play these instruments is beneficial. However, such prior knowledge will be confusing when the user attempts to play a virtual instrument which is modelled differently to its real-life version.

6.2 Interface implementation

Effects from the glove itself can affect the functioning of the system. As an example, the glove device operates continuously, and asynchronously of the host computer. To wait for new data to be transmitted from the glove device may delay the host system unnecessarily, so a polled, interrupt-driven input driver is used. This raises the problem of what to do during reported samples (see Figure 6.1). If the users' hand position is sampled at point A and some period of time later at point B, what should be done if the hand passed through a virtual instrument in travelling from point A to point B? It is likely that the user intended to 'trigger' the virtual instrument (i.e. play a note on the instrument) by making this action, and in such a situation the issue becomes important of whether to use individual sampled positions by themselves as instrument triggers (is the sample at point B inside an instrument?), or to use some function of a number of positions (has the user probably intersected the instrument in moving from point A to point B?).

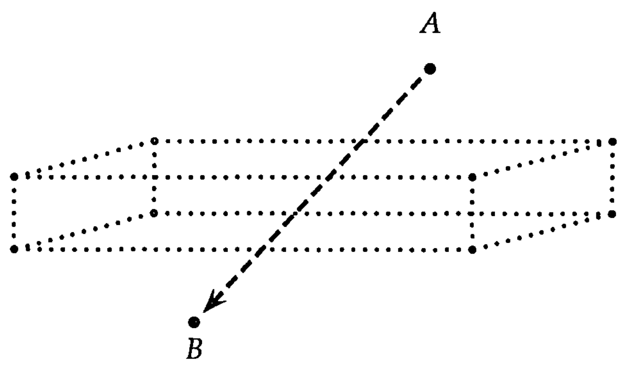


Figure 6.1: Movement through a virtual instrument

The glove device may be changed in the future, for a more robust model or one that can report higher quality data. Such a change may affect the functioning of the translation system, or the triggering of instruments, because of a difference in the units reported by the new glove, or an increase or decrease in its sample rate or degrees of freedom. The translation software ideally should be designed so that gestures for triggering instruments do not differ when performed using different glove devices, so that changes to the software are minimal when a different input device is used.

Each of these problems needed to be considered when developing the glove-information to MIDI-data translator.

6.3 The Translator

The translator has the function of reading in glove data, manipulating it in some way according to internal models of some real instruments, and sending the output to an appropriate sound generator. The translator system is arranged in three parts:

- an input section that handles data from a glove (or other input device), converting it to a form suitable for manipulation by the translation section;
- the translation section, which takes glove-data from the input section and generates output based on various tests and models of instruments. The translator is basically a *mapping system*, that maps position, orientation, finger flex and motion data sequences onto MIDI command sequences;
- an output section, which takes the MIDI commands generated by the translator and feeds them to an output device, capable of rendering sounds based on the MIDI commands it receives.

The input section has to handle many of the glove-related problems presented in Section 6.2. The main difficulty it has to deal with is the difference in physical sizes of the users of the system, most importantly in their hand dimensions. Visitors to the science centre are aged from around 2 to 3 years old up to 60 years and over. Hand sizes for this group of people vary greatly, and while a (small) range of glove sizes are available, the variation in hand sizes creates a problem for the translator section.

Translation of glove data into MIDI data requires models of the instruments to be simulated. Two different types of models were tested in development of the system, one based on the behaviour of the instruments, and the other based on how the instruments are operated. The output section is a simple interface between the translator and a MIDI device.

6.3.1 Dealing with hand sizes

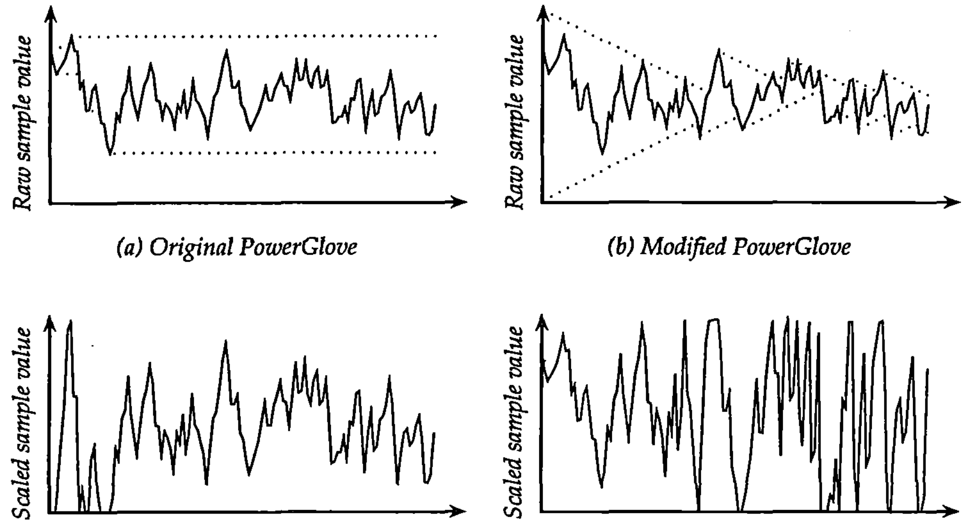
Various gestures are recognised by the translator, and used for triggering instruments (this process is covered in Section 6.3.8). All of these gestures require flexion of the fingers so the system needs to have some way of determining whether the user is flexing a finger or if the finger is flat, irrelevant of the user's hand dimensions. This becomes difficult when there is a variation in the hand sizes of users because some users cannot bend the flexion sensors to the same degree as other users, and yet the system must still be able to recognise that a finger is being flexed. On the original Nintendo PowerGlove, the initial glove used for the system, a crude form of compensation was available.

After the PowerGlove had been switched on, a user flexed a finger to angle θ, the glove would 'remember' that this was the maximum amount of flex which that finger had been subjected to, and the range of finger flexes would be between 'flat' and angle θ. If the user then flexed the same finger to angle ϕ, where $\theta < \phi$, the PowerGlove would set angle ϕ to be the new maximum, and measurements would be between 'flat' and ϕ. This was satisfactory, until another user with a smaller hand started using the glove, and could not flex the finger to angle ϕ. The maximum flexion value would stay at angle ϕ until the PowerGlove was switched off. This meant that the second user could not register a fully flexed finger with the system, and so many of the instruments would be untriggerable.

As the original PowerGlove does not report a true flexion reading, but instead reports a pre-scaled flexion value (see Chapter 7 for more details on the PowerGlove's functioning), it is not possible to determine how far a finger is actually flexed. After modifying the PowerGlove to report actual sensor readings rather than pre-scaled values, the input system was altered so that it would adapt to different users's hand sizes and flexion capabilities. Samples from flexion sensors in the modified PowerGlove are reported in an 8 bit number, which is a representation of the amount of resistance in the sensor for that finger. Due to various manufacturing defects and drift in the sensor, the range of this value differs from sensor to sensor, and even on the same sensor at different times, and some resistance is always present (flexing a sensor in one direction increases its resistance, flexing it in the opposite direction lowers its resistance, but the resistance will not go below a certain level irrelevant of how much flexion the sensor is subjected to).

To determine the degree of flexion a finger is undergoing requires calibration of the software that is to read the sensor values from the glove device.

Many laboratory systems for commercial glove devices have a form of gesture recognition software for use with the system, and require calibration of the input software before use [Stu91]. Such calibration is done semi-automatically and continuously in the PowerGlove as mentioned above. However, these systems typically have a calibration and possibly a gesture training period only at the beginning of an experimental run. Sturman [Stu91] discusses the use of continual calibration rather than a single calibration



(a) Original PowerGlove   *(b) Modified PowerGlove*

(c) Original PowerGlove   *(d) Modified PowerGlove*

Figure 6.2: Finger flexion scaling on PowerGlove and modified PowerGlove

period whereby the system keeps track of the maximum and minimum values as the experimental session proceeds, adjusting the scale of reported values accordingly [Stu91]. This method is analogous to the functioning of the PowerGlove.

Such a system is not sufficient for a science centre exhibit, as the system must be able to calibrate itself continuously to a wide range of users, as well as continuously adjust to the current user. Typically in a laboratory environment only one user operates the glove during an experimental session. For such cases Sturman's method is adequate, but in the science centre environment the system must be able to adjust to potentially hundreds of users during the course of a day.

To achieve this, the modified PowerGlove sends raw (i.e. unprocessed) finger sensor flex readings to the host, which utilises these values to calibrate the glove. Maximum and minimum reported samples from the sensors are recorded, and the range between these two values is normalized to 8 bits. A linear interpolation is used in the normalization process, as this is quick to calculate, and is adequate for the gestures used in the "Virtual Maestro" exhibit. Sturman (ibid) has found linear interpolation sufficient in many cases on a number of glove devices, as an accurate angle measurement is not usually required, but repeatability and dynamic range of the sensors is a requirement.

The maximum and minimum values are adjusted over time, decaying away from their end of the flexion range and towards the opposite end of the range (i.e. the maximum flexion value is decreased slowly towards 0°). If the current reported sensor sample is outside the minimum-maximum range, the appropriate measurement is reset to the current sample's value, and decaying continues (see Figure 6.2).

This increases the sensitivity over time to small movements, and ignores bias and offsets in the sensors. This method also allows the system to adjust as the sensors' resistances change, assuming that the resistance is still dependent on the flex that the sensor is subjected to. The original PowerGlove is unsuitable for this autocalibration scheme because the process requires raw sensor readings, and the original PowerGlove processes these sensor readings itself before passing a 'calibrated' value to the host. The input section takes raw flexion data values from the modified PowerGlove and 'stretches' them to increase the sensitivity of the sensors and make the glove device more user-independent; an important feature when a large number of users will be operating the system.

Sturman (ibid) mentions that sensor noise causing spikes in the sensor measurements can affect the continuous calibration system he describes. The calibration method used in the "Virtual Maestro" exhibit has a degree of immunity to such noise, due to the decay process applied to the range. Effects of any spikes in sensor readings are gradually removed from the system.

6.3.2 Internal data format

To cope with a number of different glove and other input devices that might be connected to the system, a common data format is used within the translator. Incoming data from the input section is normalized to this generic data format to maintain device independence within the translator. The data format for each sample includes fields for position, orientation and finger status, and is shown in Figure 6.3. The data format provides information on six degrees of freedom in the movement of the input device, as well as crude finger flexion information. Units are millimetres for positional values, and degrees for orientations. Finger flexion is simply a level between 0 and 255, that denotes the amount of flex a finger is undergoing with 0 representing flat, and 255 representing fully flexed.

Most input devices don't provide all of this information, so for these devices the unprovided inputs can be set to some arbitrary value or possibly tied to another input, as is the case for the little finger of both the original PowerGlove and the modified PowerGlove. There is no sensor in the original PowerGlove to detect flexion of the little finger, and in some gestures a fist is required (i.e. all fingers need to be flexed for the

```
typedef struct {
        int     x, y, z;
        int     yaw, pitch, roll;
        char    fingers[5];
        } glove_data;
```

Figure 6.3: C data structure for representation of input device data

gesture to be recognised as a fist), whereas other gestures require the fingers to be flat for the majority of the time, and flexed for triggering a note. Using an arbitrary value for the little finger's flexion is unsuitable as the choice of flexed or flat values will affect the gestures that can be performed. To allow the user to perform fist and flat gestures, the little finger is tied to the ring finger, mimicking its sensor value. For the gestures used in the exhibit this provides an adequate solution.

Other degrees of freedom that are not reported by the original PowerGlove include yaw and pitch information. While the 2-transmitter configuration of the original PowerGlove is capable of determining yaw, it does not report it. A 2-transmitter configuration is not capable of determining pitch, so this could not be reported. The modified PowerGlove can report yaw information to approximately ±40°. Angles higher than this are not reliably reported due to the directional nature of the ultrasonic transducers. At reasonably large angles away from a direct emitter-to-sensor path, the signal strength drops dramatically, making determination of the angle difficult (see Figure 3.5). For playing the virtual instruments, roll is the only orientation used, as this is common across all of the instruments.

The original PowerGlove's data format differs from the host's internal data format, and data values reported from it need to be scaled appropriately to match the internal format. Finger flexion data is a single 8-bit byte, consisting of two bits per finger. This gives four possible values which are reported for flexion of a finger. This is scaled to the range used in the data structure and reported from the modified PowerGlove. Position and orientation values from the original PowerGlove are also scaled to match the data structure's units. The original PowerGlove's units of position are approximately 3mm per unit along the *X* and *Y* axes, and approximately 14mm along the *Z* axis. Orientation is in the range of 0–11 30° steps. Other input devices will probably need to be scaled appropriately to match the units of the internal data structure.

6.3.3 Position extrapolation

After normalising input samples and converting them to the system's internal data format, the samples are available to be passed to the translation code. The rate at which these samples become available, known as the update rate (see Section 3.3), can be very low for some input devices. The original PowerGlove has an update rate of approximately 12–15Hz, for example. Such a low rate of computed positions is acceptable in some applications, but for realistic real-time musical performance this rate is too low. The system response lags behind the user's actions so dramatically that the direct cause and effect relationship of musical performance is practically lost. Users of the system attempting to play it find the lag distracting, almost to the point of the system being unusable. Rapid movements are often ignored by the system, as they occur too quickly for it to register them (this is a form of aliasing, a problem common in digital signal processing), and the system gives the appearance that it is malfunctioning. Some solutions to this problem are to increase the sample rate, which has been done on the modified PowerGlove, and to change the way instruments are triggered (see Section 6.3.4).

Another problem is also present with low and medium sample rates; that of the asynchronous operating nature of the input device. Position samples are calculated at approximately regular intervals, but the gesture recognition software can take varying amounts of time to process the incoming data, and the cycle rate of the gesture recognition software — the time between successive requests for position data; about 3–5msec on a fast 486 PC — and the update rate of the input device are unlikely to be the same (original PowerGlove: between 60 and 80msec; modified PowerGlove: about 24msec). The modified PowerGlove uses a serial interface to report data to the host system, and the reception of a command to send new position information, the processing of the command and the sending of the requested information can take a significant period of time. To allow the gesture recognition software to run at full speed the input driver is interrupt driven, and requests from the gesture recognition code to the input driver for new position samples are performed by polling the input driver for new data. By this method samples are obtained when the gesture recognition code is ready to analyse them, rather than when a new sample becomes available. This allows the gesture recognition code to operate efficiently, with very short waiting periods when it attempts to obtain data.

Because of the asynchronous nature of this form of operation, the situation can arise when the gesture recognition software requests a position sample in-between two positions reported from the input driver, as in Figure 6.4. There are a number of different positions that could be reported by the input driver in this situation. The simplest and fastest position value to report is the most recent position sample, at position A. However for gesture recognition by feature analysis (discussed in Section 6.3.6), determination of the glove's velocity is disrupted by duplicating previous samples. This occurs because the velocity of the moving glove is determined by the distance travelled since the most recent sample, divided by the time that has elapsed since that sample was taken. If the sample is duplicated because a current sample is unavailable, the glove's apparent velocity will be zero. A sequence of samples over time under this scheme generates a stream of velocity readings indicating a movement followed by a stationary period (see Figure 6.5), when the input device may have been in constant motion during this period.

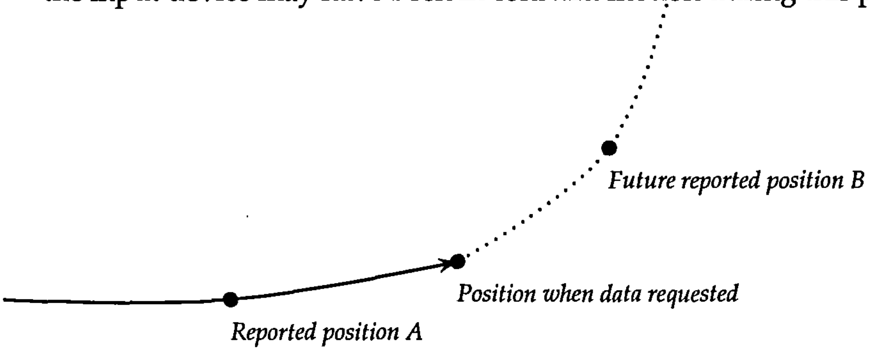


Figure 6.4: Position data request during interval between samples reported

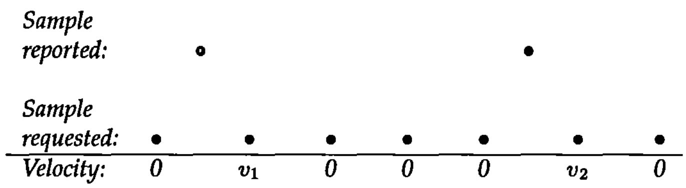
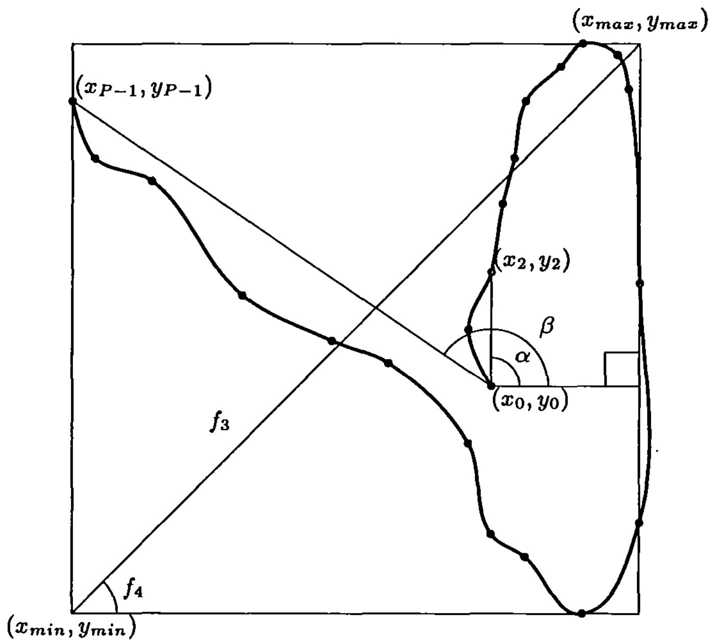


Figure 6.5: Velocities reported with previous sample repeated

The gesture recognition software uses the difference between the previous reported sample and the current reported sample (which is a duplicate of the previous reported sample in this case), to determine what direction the glove has moved in. This requires a continuous stream of velocities rather than a 'burst' velocity followed by a stationary period. To cope with this situation a linear extrapolation method is used on two previous sampled positions in the input driver, so that requests for positions made by the gesture recognition code are passed estimates of the glove's position whenever an actual sample is unavailable. This gives a continuous stream of velocities and allows the gesture recognition software to be flexible as a number of other features about the user's motion are also available. An almost arbitrary level of recognition can be achieved, from basic directions such as "up", "left" and so on, to recognition of a path constructed from a number of samples, such as the gesture in Figure 6.6. Recognition of such gestures using feature analysis requires a number of measurements to be made over a period of time and using more than one position sample. Such measurements would not be calculable if samples were simply duplicated. A more complex extrapolation method is probably a better choice for applications requiring high accuracy and low noise, as sharp movements can introduce error when a sample is extrapolated just before a new sample is reported (see Figure 6.7). This effect is utilised in the *linear prediction* compression scheme, where the difference between actual samples and predicted values is used to obtain compression [BCW90]. A better prediction of the next value reduces the error and increases the compression. For speech signals, ten or more previous samples are commonly used to predict the next value.



See [Rub91] for a description of the features labled in the above gesture.

Figure 6.6: A complex gesture over a number of samples

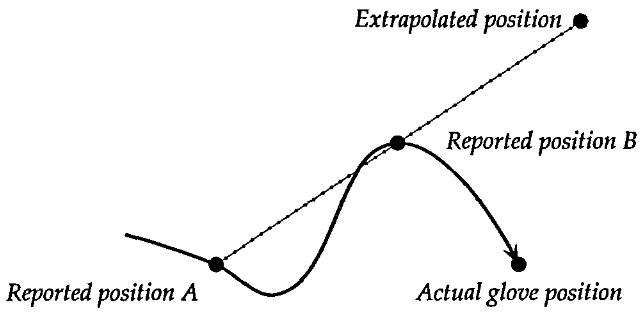


Figure 6.7: Errors in linear extrapolation of positions (2 dimensions)

6.3.4 Mapping software

The translation section takes glove data from the input section and transforms it to output data according to how it maps onto the modelled instruments of the system. In the first prototype, these instruments consisted of a few simple three dimensional models of real instruments, and some abstract three dimensional shapes that had audio events associated with them. Instruments represented were a piano-styled object with black and white 'notes', represented as a rectangular box with behaviour similar to that of a piano, and 'proximity' objects that would play a note if the user's hand moved within a certain distance (the *trigger distance*) of the instrument. The pitch of the played note, its amplitude, or both, were variable if the user moved their hand around while within the trigger distance of the proximity instrument. These proximity instruments are similar in function to the Theremin discussed in Section 4.2.

Triggering of the piano-style instrument is based on an intersection of the user's hand motion with the instrument's rectangular box volume. Using inclusion tests only on reported positions from the original input system is not sufficient to determine whether the user has interacted with an instrument or not, as the sample rate of the original PowerGlove is approximately 12–15Hz, well below what is required for musical performance. In Figure 6.1, the user's hand passes through the virtual instrument, but neither of the points are enclosed by the instrument, and using a test for inclusion of a position sample by an instrument gives incorrect results. To interact correctly with an instrument we need to know if the user attempted to play the instrument in the time between the last position sample and the current one, rather than whether the current position is interacting with the instrument. For a piano, this means "Has the user's hand passed into or through the virtual piano, in a downward direction since the last sample?". This is determined by the intersection of the path of motion of the user's hand and the instrument, and the direction of that motion. Satisfactory results are achieved with the assumption of a linear path from point A to point B, and a linear estimation of the path is simple and fast.

For the virtual piano instrument, motion downwards, into or through the instrument is used to trigger a note. The pitch of each note is determined by where the path of the user's hand motion intersects the major axis of the modelled piano's rectangular box shape. The initial amplitude of the note is proportional to the length of the path, because a long path length is achieved by movement at a high velocity, which implies the user's intent was to strike the virtual piano with some force causing a loud note to be played.

Because of variations in users' heights, users found the three dimensional modelled instruments difficult to locate in space. The instruments were either too low, too high or too wide. Positions reported from the original PowerGlove gave X, Y and Z values in the range from -125 to $+125$ units of the 'origin', or about ± 375mm from the origin. Positions beyond this distance are clipped to the maximum deviation. This operating volume can be moved about by pressing the "CENTER" button on the original PowerGlove's keypad. Doing so makes the glove's current position the centre of the volume, or the new centre. If a tall user sets the origin (by pressing the "CENTER" button), to be reasonably high, any subsequent shorter users unaware of the high origin will find it difficult to play any instruments as the position of their hand will be outside the range of the PowerGlove's reported values, and unable to trigger instruments. If these shorter users are also unaware of the function of the "CENTER" button then it is likely that they will not succeed in operating the system. On the modified PowerGlove the "CENTER" button is not available, and so some other form of reset required.

Providing a gesture or button for manual resetting of the system requires that some instructions are provided for the user to be aware of how to reset the system. In the case of a gesture, the original PowerGlove may clip position values for that as well, interfering with the gesture and possibly making it useless. If for example the user is short, and the origin is high enough for glove positions to be clipped even when the user stretches their hand as high as possible, the software will not be able to determine that they have made a gesture for moving the origin, if the gesture involves any upwards or downwards motion. For a button (or any other manual reset trigger) instructions are required on when and how to reset the system. Adding instructions of any sort on how to operate the system tends to move away from the goal of virtual reality being a *natural*, intuitive interface; instructions on how to use the interface should be avoided if possible. It was observed that users rarely read the instructions provided, and attempted to operate the system as they thought it should be operated. Having a manually operated but not user-accessible reset (i.e. one that an exhibit floor demonstrator would operate, rather than the visitor using the system), is infeasible, because of the high expense of having a demonstrator available to reset the system for every user. The science centre environment aims at having few demonstrators on the exhibit floor, partially because of the cost of employing them, and partially because the exhibits are for the visitors to explore. Any exhibit-related quirks such as resetting between users is ideally performed either by the exhibit or the users themselves. Demonstrators serve the primary purpose of explaining the underlying principles of the exhibit if required, and fixing major problems with exhibit functioning.

Therefore, some form of automatic reset or adjustment is required, to allow the system to function autonomously. The modified PowerGlove, presented in Chapter 7 returns absolute positions and a *drifting origin* can be used to centre the virtual world around the average position of the glove. In this process a number of position samples over time are used to calculate a moving average of the glove's position. This is subtracted from the current sample to give a moving origin that automatically positions itself in the most reachable position. If a short user tries the system after a tall user has been using it, the system will effectively move the instruments down to within the new user's vertical reach. This method can suffer from the user's hand pausing at an unusual height, which would move the origin towards that position until the user moved their hand again, but the system would continue adapting as soon as they move. However, the apparent change in the instruments' positions may be disturbing to the user. The time for this readaptation is arbitrary, but should be chosen carefully — adjusting too fast 'nullifies' the user's movements, and adjusting too slowly defeats the purpose of the technique, because the origin would appear to be stationary.

The original PowerGlove is unsuitable for this method of automatic adjustment because an internal model of its position is maintained by the CPU controlling the glove, and there is no access to this from the host machine. It is from this internal model that the CPU determines what the glove's position is within the $\pm$375mm range of the movable origin. Positions from the glove's CPU could potentially be clipped, and so the drifting origin model in the host cannot work to its full potential. The origin of the glove can be moved, but not under control of the host. However, if the host kept track of the glove's positions and mapped them onto its own internal real-world model, adjusting positions by an offset that changed whenever the "CENTER" button was pressed, it could keep track of the glove and increase its range artificially. This requires hardware changes to the glove, but does not improve its data quality, and cumulative errors from mapping new glove positions to the host's model makes the drifting origin process of little benefit when used with the original PowerGlove.

6.3.5 Position jittering and hysteresis

Another problem to deal with is that of low level noise in the input device, that causes 'jittering' between adjacent values. This happens when a glove is positioned near the boundary of two positions, as in Figure 6.8. If fingers of the glove are modelled as lines with no width, then they can effectively trigger both note A and B repeatedly, as low level noise will cause the position of the finger to be apparently shifting back and forth between the two notes.

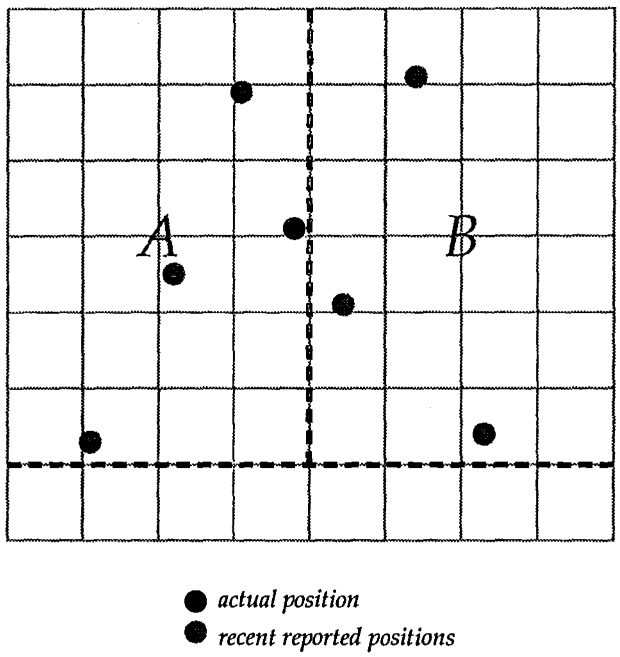


Figure 6.8: Jittering caused by incorrect modelling

This effect is caused by the model of the finger having no width. In reality human fingers have a finite width and curved shape, and when playing a piano keyboard for example, the finger must move a considerable distance off a note before it will cease playing. Because of the height difference in a key that is being played, and therefore pressed down, and an adjacent one that is not being played, the user's finger must move some distance onto the adjacent key to depress it and play the note. The finger must also move almost entirely off the original key to stop it playing, so while it is reasonably easy to have two keys playing at once, it is difficult to rapidly oscillate between them, ceasing one note's playing and activating the other, as in the case of jittering in the position of the input device. Some solutions to this are to implement some form of hysteresis on the notes, to model the fingers more realistically, and to average the most recent position values so that small deviations are removed. This last method has been used as it is the simplest and fastest to compute. It does have the disadvantage of reducing the effective sample rate of the system, because rapid movements are cancelled out (although this depends on the averaging routine used), and it does not entirely remove jittering, but merely lessens the effect to an acceptable level. A clipping algorithm is also applied to the position values, to the lower few bits of each position. This reduces the accuracy but also reduces the jittering. It does not affect the sample rate.

Hysteresis is more difficult to implement in the system because this process requires the system to keep track of which finger triggered a note, and where the finger is now, to enable the system to determine what note(s) to play or stop playing. If the user deactivated the note (by lifting their finger off the virtual piano keyboard, for example), moved their finger to an adjacent note, or managed to activate an adjacent note without deactivating the current one, the system would need to determine which notes to play or release, which on the initial instruments used in the prototype, was difficult to achieve and added a large amount of complexity to the system.

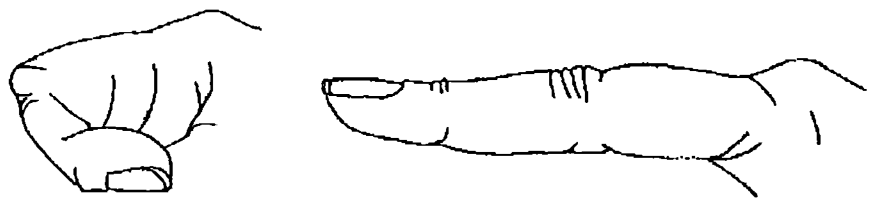
Modelling the fingers more accurately is a good approach, but as with the hysteresis process this method requires increasing the complexity of the system by a large amount, and was not chosen as a result. The system needs to be small and efficient, in case the host machine should change to one of lower capabilities, as could be expected if the exhibit's cost was to be reduced.

6.3.6 Posture and gesture recognition

To cope with the varying physical differences in users of the final version of the "Virtual Maestro" exhibit, a gesture recognition system is used, instead of the prototype three-dimensional model approach described in Section 6.3.4. This has the benefit of being user independent, in that heights of the users are not critical to the application's operation. Gestures are recognised on a relative basis from where the last position sample or samples were taken. Sturman (ibid) breaks gesture recognition into two sections, that of *gestures*, or movements made by the input device in three-dimensional space, and *postures*, the status of the fingers and hand while making the gesture.

The gesture part of the recognition system is based on work by Rubine [Rub91] and Sturman (ibid). Statistical features of the input stream of positions are analysed to determine a gesture. Such features include the initial starting angle of the gesture, its total path length, the duration, and the maximum speed attained. These features are then evaluated according to some function, that matches stored gestures' features to the features of the current gesture, and classifies the current gesture according to which stored gesture had the most similar features.

The feature recognition software used in the "Virtual Maestro" exhibit only analyses simple features of the input data, such as the directions the input device is moving in, and how rapidly it is moving in those directions. As most musical instruments utilise simple postures in the act of triggering the instrument, typically that of a finger being flexed or not, the postures of the hand are analysed in a similar way to the gestures, with fingers being either in a *BENT* or *FLAT* state, corresponding to the two positions presented in Figure 6.9.



(a) *BENT* finger posture

(b) *FLAT* finger posture

Figure 6.9: *BENT* and *FLAT* finger postures

Because of the effects of the decaying finger flexion ranges, discussed in Section 6.3.1, a finger's flexion will tend towards either BENT or FLAT after a short period of time, because the two ends of the range will converge towards the current sample, reducing the relative distance between the sample and the nearest range limit (see Figure 6.2). As a result, the posture of the user's fingers will tend to a combination of BENT and FLAT positions under most circumstances.

Orientation of the user's hand is also used in the determination of gestures, and these are broken up into 90° regions, being *HORIZ* when the hand is flat, palm facing downwards, and *VOUT* when the hand is tilted 90° clockwise (i.e. the palm is facing to the left, and the back of the user's hand is facing to their right). The orientation applies to the user's right hand, as both the original PowerGlove and the modified PowerGlove (the two glove-based input devices used in the system) are right-handed.

To determine whether an instrument is triggered, a combination of movements, orientation and postures of the input device are tested against a number of predefined movements, orientations and postures, corresponding to approximate postures and gestures required to play a particular instrument. As noted in Sections 6.1 and 5.2, some instruments are 'altered' so as to be playable with a single glove.

6.3.7 Instruments

Currently the "Virtual Maestro" exhibit simulates 'piano', 'drums', 'strings', 'choir', and 'saxophone' virtual instruments. The instruments are 'triggered', to initiate the transmission of a MIDI 'NOTE ON' command, by various gestures and postures that represent the instruments being modelled (see Figure 6.10).

Having a number of different instruments simulated requires some method of selecting each one to play it. This selection could be accomplished in a number of ways, some of which are:

- natural gestures according to the instrument where the motions of playing a piano selects the 'piano' instrument, the motions for playing a drum set select the 'drums' instrument and so on;
- a menu-selection styled gesture for picking an instrument from a pull down list;
- different buttons on the exhibit to select an instrument;
- automatic selection via a sequence, where one instrument is selected for some period of time, then another is selected, and so on down the list of available instruments, cycling back to the first.

In trying to keep usage of the interface as intuitive as possible in accordance with the trends towards transparent interfaces, the natural gestures approach is used. These gestures could be made more realistic (for example, the thumb could also have an effect in the 'piano' instrument) if a different method was used to select instruments, but using natural gestures allows users of the system to begin playing any of the instruments supported, and the system switches to that instrument immediately. To achieve immediate switching to an instrument, the gesture performed had to be unique over a very short period of time. To this end most of the instruments require a reduction in their capabilities so that the triggering gestures can be differentiated. An example of this is the requirement for the 'piano' gesture to have a flat thumb, because the 'strings' gesture is the same orientation, and finger flexion from the other fingers could be confused, but the thumb and forefinger must be flexed on the 'strings' instrument as if holding a bow, which differentiates the two gestures.

Gestures used in the model for the 'piano' instrument are a downward slowing movement such as that made when hitting a piano keyboard with the hand, or the flexion of a finger irrelevant of the hand's motion. Both of these two actions are capable of triggering the 'piano' instrument if the user's hand is in the 'HORIZ' orientation, and their thumb is 'FLAT' (i.e. not flexed). If the previous gesture was one of the above and the user has moved left or right, then the current gesture is also classed as a piano trigger, irrelevant of whether the current gesture is downward slowing motion or a finger is bent. This is to cover the case of a note or notes being played and the user moving their hand left or right, as would be the case when they 'slide' up the keyboard. The pitch of the played note is controlled by the X axis position of the user's hand.

The 'piano' instrument modelled above does not respond correctly to releasing one or more keys by lifting the hand away from the keyboard when some of the user's fingers are flexed. This aspect of the 'piano' instrument's behaviour could possibly be removed by using a different model of the finger motion. Such a model might attempt to capture whether the finger had been flexed or straightened rather than if it was currently flexed or straight. This is similar to a number of computer systems that use mice as input devices, where *Mouse-Down* and *Mouse-Up* events occur when a mouse button is pressed or released respectively.

In the case of the 'piano' instrument such events would be *Finger-Flexed* and *Finger-Flat*, and upwards movements of the user's hand would generate Finger-Flat events for all fingers. Similarly if the user moves their hand downwards, Finger-Down events would be triggered for each finger that is flexed, or for all fingers if none are flexed. This model of the 'piano' instrument may be added to the system at a future date.

The 'drums' instrument is played by making the motions of playing a drum kit. The motions used are a clenched fist posture as if holding a drumstick, while the hand is in the 'VOUT' orientation, with forward and downward motions of the hand as would be done in playing a drum kit. Different drums and percussion instruments are spread out in the *X* and *Y* axes, with wrap-around in both axes to allow people of many different heights to play each instrument. No concern was given to layout of drums, or types of drums used, in real drum kits. Instead a variety of drums and other percussion instruments was used to provide an interesting system for novice users. Such users are generally unaware of the layout of a drum kit, so would gain little benefit from a realistic layout of the drums and percussion instruments.

These instruments include some which are normally activated by foot pedals as well as those activated by being struck with a drumstick. Due to an initial mistake in the gesture recognition code for the 'drums' instrument, triggering was done repeatedly by the motions described above. In other words, a motion forwards or downwards caused a group of 'NOTE ON' MIDI commands for the duration of the motion, instead of a single 'NOTE ON' when the motion ceases as would be the case when a drummer struck a drum with the drumstick. These 'rapid-fire' drums were retained in the exhibit because they enhanced the 'drums' instrument, making it a more interesting instrument for novices to play. Being able to produce a rapid drum roll was a novelty such users were unlikely to be able to achieve with a conventional drum kit. The effect acts on all of the percussion instruments mapped onto the 'drums' instrument. Some of the most interesting percussion instruments this rapid-fire system affects, are crash cymbals, which can sound very impressive at high speed, and the high and low tom-toms and snare drum, which are good for drum rolls.

The 'saxophone' instrument is triggered by gestures common to the 'drums' and 'piano' instruments. Pitch is triggered by fingers being flexed (not including the thumb) as for the piano, and the hand must be in a 'VOUT' orientation similar to the 'drums' instrument. In other words, the hand must be tilted sideways, with the palm facing to the left, thumb extended (i.e. flat) and fingers flat. To play one or more notes, the fingers are flexed.

The user's thumb must not be flexed when playing the saxophone because under certain circumstances this posture makes it identical to that required for the 'drums' instrument. Pitch of played notes is controlled by the Y axis value of the user's hand. This avoids the requirement of operation of the keys on a real saxophone. However, the pitch on a real saxophone is determined by a combination of keys pressed that decode a particular note. A saxophone is therefore monophonic by nature, since all of the player's fingers are required to decode a note from the range of available notes. The 'saxophone' instrument does not have this requirement, and can produce a note when any finger is flexed. It can produce more than one note at a time, making it a polyphonic version of its real-life counterpart. In Section 6.3.8 the 'saxophone' instrument is discussed again in reference to its polyphonic nature. The saxophone is a example of a two-input instrument where the fingers control the pitch, and triggering is controlled by breath of the player.

In the "Virtual Maestro" exhibit these inputs have been altered as there is no physical ability in the system to use breath for a trigger, mostly for hygienic reasons (many users blowing into a device to control amplitude — with more than 2000 users passing through the science centre within two days of opening, a breath controller would become home to many diseases in a short time), and pitch information can be determined by other methods. Consequently the 'saxophone' instrument is different to its real life counterpart.

String instruments are normally two input instruments, requiring fingers to control the pitch of notes and a bow to trigger notes to play. To include such an instrument in the "Virtual Maestro" exhibit requires mapping of one or other of the required inputs to an available degree of freedom of the glove input device. The 'strings' instrument uses a bow posture in the 'HORIZ' orientation, with the thumb and index finger flexed while all other fingers are flat, combined with left or right motion to trigger notes, and pitch of the notes is controlled by the Z axis position of the user's hand. This is a similar gesture to that of playing a real string instrument.

A 'choir' instrument was included as a novelty instrument for users to "conduct". To control the choir a pointing posture — extended index finger, flexed thumb and remaining fingers — is made while the hand is in the 'HORIZ' orientation. This triggers the choir to produce a single note with pitch based on the X axis position of the user's hand, and amplitude based on the Y axis position.

6.3.8 Instrument modelling

Each instrument modelled by the system is 'triggered' to activate one or more notes and cause the appropriate 'NOTE ON' commands to be sent to the output section. For the prototype three-dimensional models, keyboard style instruments consisted of a series of rectangular boxes arranged side by side in a line, each corresponding to a consecutive note in the C-major scale. If the user's hand passed into or through a box the pitch value associated with that box was sent to the MIDI output section to be incorporated into a 'NOTE ON' command. Proximity instruments sent a single pitch to the output section, then followed with a continuous stream of values to be used as MIDI 'PITCH WHEEL' parameters. In the gesture based system, instruments have a number of *note regions* where triggering the instrument will cause a particular note to be played. To cope with differences in users' heights, discussed in Section 6.2, the instruments wrap around on themselves repeating a finite number of notes at different intervals. This potentially allows any user to play every note provided on an instrument, irrespective of their height.

The repeated range of playable notes on all virtual instruments except the 'drums' instrument is one or two octaves of white notes (i.e. in the key of C), although other scales could be used, such as those discussed in [Deu92]. For the 'drums' instrument, various specific notes are used, which correspond to certain percussion instruments on the percussion channel of the synthesiser (many modern synthesisers have a channel exclusively for percussion instruments).

The instruments use octaves ranging either from 'middle C' (MIDI note number 60) to one octave above middle C (MIDI note 72), or from one octave below middle C (MIDI note 48) to one octave above. Only a limited range of one or two octaves is used because although the MIDI standard allows for 128 notes ($10\frac{2}{3}$ octaves) on a channel, if the entire range were included some notes at the extremes of the range would be out of the reach of some users. The 'piano' instrument uses 10cm wide note regions so to use all 128 notes of a channel as defined in the MIDI standard would make the piano's keyboard over 10 metres in width. Another reason for not using the entire range of MIDI notes is that some synthesisers don't support notes at the extreme ends of the range and the range of the supported notes varies depending on the instrument in use, but the middle few octaves are common on all instruments.

A limited range of chords are possible current version of the "Virtual Maestro" exhibit. Because of the lack of abduction sensing (lateral movement of the fingers) in the glove device used, many chords cannot be performed if consecutive fingers are mapped to consecutive notes. The system could possibly be configured to use an individual finger flexed as meaning a single note, and multiple fingers flexed to be mapped to a particular chord. On some low end electronic synthesisers, an option is available to allow the player to press a single key on the lower end of the keyboard, and the synthesiser will produce a given chord. In this way the synthesiser can appear more flexible in the sounds it can produce, but this is achieved at the expense of losing the semi-realism of playing the notes that those keys represent normally. This kind of capability could be added to the system, with a group of fingers flexed decoding a particular chord to play, but on the current version of the exhibit fingers are mapped to adjacent independent notes, to maintain consistency with most real-life instruments.

This gives four consecutive notes playable simultaneously for the 'piano' and 'strings' instruments and four saxophone instruments on consecutive pitches. There are only four because the thumb is used to determine whether the 'piano' or the 'strings' instruments are being triggered, and whether the 'saxophone' or 'drums' are triggered. Also because the little finger mimics the ring finger's flexion value when both the original PowerGlove and modified PowerGlove are being used, two adjacent notes will sound when the ring finger is flexed.

The 'saxophone', 'choir' and 'strings' instruments originally used continuous pitch control (the pitch could vary continuously instead of in discrete tonal steps), via the MIDI 'PITCH WHEEL' controller command, as this was more representative of the functioning of these instruments in real life where a single device produces the entire range of notes for an instrument. In humans the vocal tract is capable of producing a range of notes, and on string instruments each string is capable of producing a continuous range of pitches. Playing two adjacent pitches notes in quick succession can have a slurred effect due to this single source of pitch generation changing from one pitch to the other while still producing sound. This contrasts with a piano, where each key has its own pitch source, and changing from one pitch to another occurs in definite steps. The 'saxophone' instrument was also initially designed to operate in this way to produce the slurring effect described above, but this was an incorrect model of the saxophone as although there is one reed to produce excitation for the sound, the pitch of played notes is controlled by varying the length of a resonant tube along which the sound passes. The length of this tube is determined by which keys are being depressed on the instrument by the player's fingers. Also while conventional saxophone and string instruments are monophonic in nature (i.e. can produce only one pitch under normal circumstances), the 'saxophone' and 'strings' instruments used in the "Virtual Maestro" exhibit are not bound by this limitation. Essentially four uniquely pitched notes can played simultaneously.

Currently, as for the 'piano', these four notes are consecutive ones, but the mapping system can easily be altered to play a predefined chord if a number of fingers are triggering the instrument. The 'saxophone' could be set up to behave like its real world counterpart (i.e. only one note produced, with the pitch encoded in which fingers are being flexed), but it was felt that doing so would limit the system, both by making it harder for novices to achieve any form of control over the virtual instrument (for a novice the virtual instrument would be as difficult to play as a real saxophone), and by not utilising the MIDI channel to its full capabilities (only one note being produced when most synthesisers are capable of polyphonic note reproduction).

Note regions are very large in most of the instruments to facilitate ease of playing. The 'piano' instrument, for example, has 10cm wide note regions, so that keys are easier to find. Because there is no visual feedback in the system, users found it difficult to locate keys distant from the one they were playing if the note regions were small, but notes were more accurately located with larger note regions.

6.3.9 Output processing

Triggered notes are inserted into a list, where they are flagged for removal on each successive pass through the simulation loop. The flag is set to indicate that the note should be removed, in case the user is no longer playing it, and cleared by the triggering software if the user is still playing the note. Notes that still have their flag set after the triggering section has been called are removed from the list, and a MIDI 'NOTE OFF' command is sent to the output driver. This potentially allows the user to trigger a number of instruments with one finger, and have each one of them handled independently. The method makes it difficult, however, to perform actions on related notes, such as hysteresis between two adjacent notes. The reason for this is that no record is kept of which finger triggered a note, and at what time. To implement the hysteresis system mentioned in Section 6.3.5, the system would need to know whether two adjacent notes had been triggered and if so, in what order were they triggered.

Some initial development of the output stage was done on an Apple Macintosh computer running an application that utilises Apple Computer Inc.'s Midi Manager Tool Set. This is a driver for a MIDI interface, and provides calls to access and manipulate MIDI applications and devices on the Macintosh [App91]. Development was done on this system to test the MIDI functionality of the host output driver, before deciding on a MIDI interface for the IBM PC compatible host machine. The Macintosh computer was used as a MIDI emulator, receiving commands from the IBM PC compatible and producing sounds, then later in development, outputting the MIDI commands to a synthesiser through an available Macintosh MIDI interface. This is in keeping with the modular design of the system, where each part of the system is designed and tested independently before they are all combined together.

6.4 Summary

In creating a virtual musical instrument simulator, a number of methods of instrument simulation exist. Some may be more suitable for a particular environment than others, depending on the application and on the devices used for input and output. This chapter has covered some of the design issues involved and presented two possible solutions for achieving a virtual instrument simulator, according to the aims and objectives presented in earlier chapters. Whichever method is used depends on what the aims of the project are, and what devices and platforms are available. The cost of the system also has an effect, and were finances available, the system would probably differ from that described in this chapter. A graphical display and the computing hardware to drive it would have steered the development towards a more physically correct system, similar to the three dimensional model-based method tried initially in this chapter, and many of the techniques developed in the gesture recognition section would probably be left out.

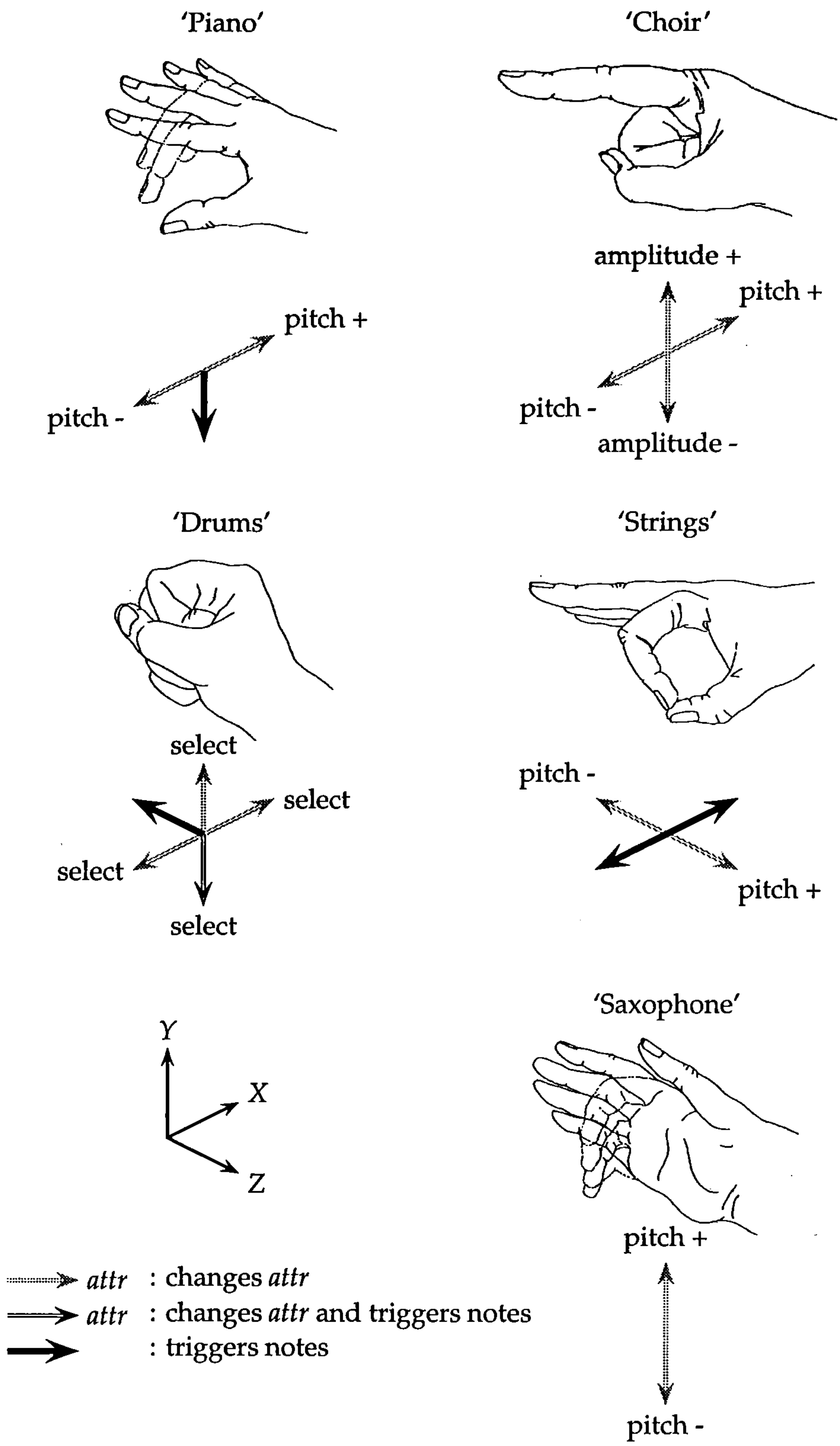


Figure 6.10: Gestures and postures required for triggering instruments

Chapter 7

Redesigning the glove device

Use of musical instruments — and indeed many other objects in the real world — requires some form of manipulation of the object in use, most often performed with our hands and fingers [AB92]. Because of this primarily hand-and-finger form of interaction with the real world, the most natural form of input device to use in a virtual reality system is one that takes hand-and-finger information and relays it to the computer to be acted upon. Such a device most often has sensors configured in a glove-like device which is worn by the user, giving the sensing system a 'direct connection' to the measurements of interest such as degrees of finger flexion.

As well as the glove fitted with sensors, other methods exist for determining hand and finger information. The most notable is that used by Myron Krueger in his Videoplace and Videodesk applications [Kru83, Kru91]. These systems rely on image analysis to determine the position and orientation of the user's hands, fingers and other limbs. Such a system is computationally expensive, requiring a large amount of image processing and pattern recognition to determine the desired information. In order to achieve the required sample rates of real time operation, these systems typically require a special purpose hardware system, as software running on general purpose computers is too slow. These hardware systems can be expensive and are not as flexible as a more software based solution. An advantage of the image analysis system however, is that no devices need be attached to the user, which improves the usability and lifetime of the system dramatically since there is much less mechanical stress on components.

Use of a sensor-fitted glove device allows for cheaper sensor technology and fewer problems with sensing, such as occlusion of tracked objects (a major concern in image analysis based systems) but it carries the penalty of a high level of mechanical stress on most of the components of the glove, often including the sensors themselves. Because of the high costs and problems associated with an image analysis system, and the availability of a cheap sensor-fitted glove device, the glove device was chosen as an input device to the "Virtual Maestro" exhibit.

7.1 The Nintendo PowerGlove

In the early 1990's a glove device known as the PowerGlove became available as an add-on system to a home video-game console, the "Nintendo Entertainment System", or NES. The PowerGlove offered another level of game play to many of the existing games and new games were written to make use of the PowerGlove's unique features. For a low price of US$100 when it was released onto the market, the PowerGlove allowed finger flexion sensing in three fingers and the thumb and position and orientation of the hand to be determined. The reason for its low cost compared with other glove devices of a similar nature, such as the US$8000 VPL DataGlove, and the Virtex CyberGlove at US$6500 + the cost of a tracking device, is due to the low cost and mass production of the finger flexion and position sensors [Rhe91]. To be marketable to the home video-game market the glove had to be cheap to produce and cheap to buy. While it was initially brought out to be used on a games console researchers quickly began interfacing it to other computers [Pau91], as the PowerGlove was a readily available product capable of reasonable quality data output. The technical specifications for the PowerGlove are presented in Table 7.1:

- Sample rate: approximately 15Hz;
- Positional accuracy: (X and Y axes) approximately 3mm; (Z axis) approximately 14mm;
- Range of positional data: approximately ±375mm from a central movable origin;
- Rotation: 0–360° in approximately 30° steps;
- Finger flexion: 2 bits per finger, adaptive (i.e. the range varies dynamically under control of the PowerGlove's CPU).

Table 7.1: Technical specifications of the Nintendo PowerGlove

This information was obtained from software used for interfacing to the PowerGlove, provided with PCVR Magazine [Gra92]. Also in this issue of the magazine, and in [Egl90], is information on how to connect a PowerGlove to an IBM-PC or compatible computer.

7.2 Functioning of the PowerGlove

The PowerGlove makes use of low cost ultrasonic transceivers in its tracking stage, to determine position and orientation. Ultrasonic pulse trains are emitted alternately from transmitters mounted on the back of the glove (see Figure 7.1). These pulses are received in one, two or all three sensors mounted in a L-shaped bar that under normal operation (i.e. with the NES) is positioned upon the user's television set. The received signal is processed and then fed to a CPU in the glove's forearm unit. The time taken for an ultrasonic pulse to travel from the transmitters to the sensors is proportional to the distance the pulse travelled through the air in that time (delays in the electrical circuitry are small, and roughly constant). The CPU determines the position of the glove by triangulation methods applied to the distances of the pulses' travel to each sensor.

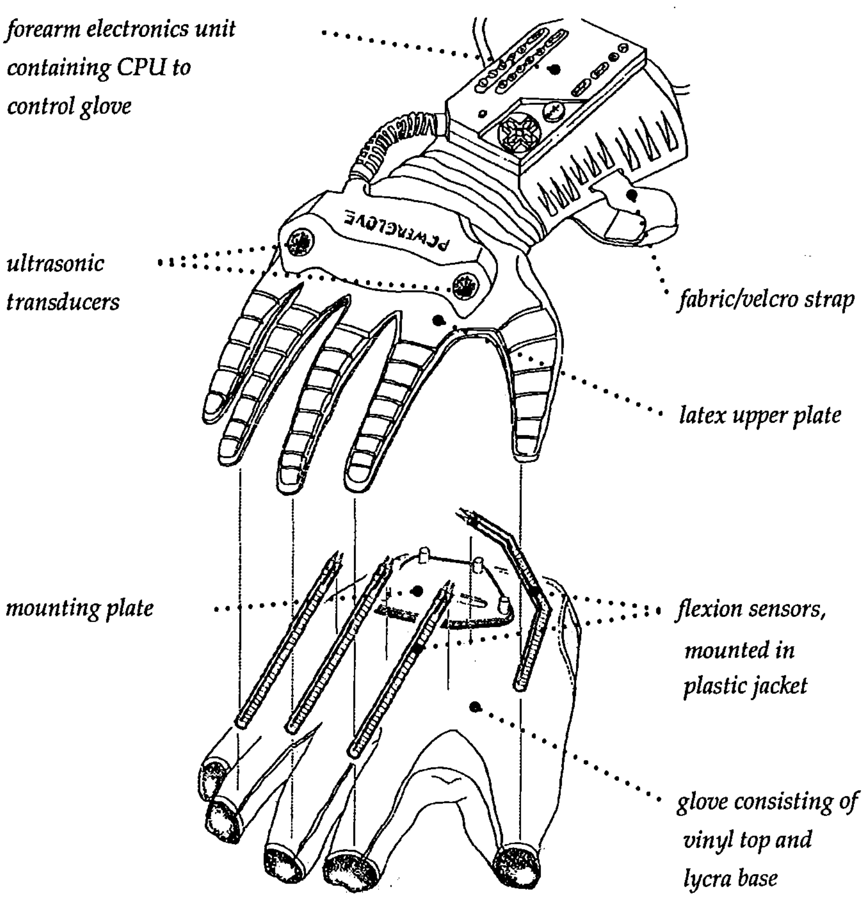


Figure 7.1: Construction of the Nintendo PowerGlove

Finger flexion is determined by the use of flexible plastic strips coated with resistive ink, which are mounted on the tops of the first three fingers and the thumb of the glove itself. These sensors are encased in the latex rubber of the glove, and are very difficult to access (doing so requires dissection of the latex shell). The resistance of the ink changes when the strip is flexed, allowing finger flexion to be determined via an analogue to digital converter in the CPU unit. Data from the flexion and position sensors is transmitted serially (not via the RS-232 protocol however) down a line to the host machine.

The PowerGlove is very similar to an earlier device known as the Z-Glove [ZLB+87] which also used ultrasonic transceivers to determine position and orientation, and optical flexion sensors to determine finger flexion. Thomas Zimmerman created the Z-Glove initially to play virtual "air-guitars" and other instruments, but the system was seen to have more widespread uses and was developed into the DataGlove [PT92]. The Z-Glove is the predecessor of VPL's DataGlove, which uses fibre-optic sensors to accurately determine finger flexion, and an electromagnetic sensor to determine position and orientation.

The PowerGlove's data quality is correspondingly less than that of the DataGlove, due to the low cost of the components and methods used for sensing data. The DataGlove, for example, can sense finger flexion on two of the three finger joints on each finger, the metacarpophalangeal (knuckle) and the proximal interphalangeal joints (adjacent to the knuckle). It can also sense the abduction, or side-to-side movement of the fingers and thumb. The EXOS Dextrous Hand Master measures all three finger joints as well as abduction of the fingers, giving a total of 30 degrees of freedom in just the joints alone (i.e. not including position and orientation of the hand itself).

The PowerGlove only senses the flexion of three fingers and the thumb, which is adequate for the "Virtual Maestro" exhibit, although some more measured degrees of freedom would allow more realistic modelling of some of the instruments, notably the piano, which would benefit from correct performance of chords. Increasing the number of finger joint measurements on the PowerGlove from one to two or three was not considered to be important because observation of the flexing of fingers showed that whenever the finger was flexed most often it was the metacarpophalangeal joint that was flexed, rather than the proximal interphalangeal joint.

Because of the poor data quality of the PowerGlove for musical performance, most notably its low sample rate, and high costs of other gloves and their fragility compared with the PowerGlove, a PowerGlove was modified and used as the input device. Much of the original PowerGlove componentry and structure was able to be re-used in the creation of a prototype modified PowerGlove. Because of the low cost of ultrasonic transducers, these were used as the tracking system on the prototype modified glove, rather than using a more expensive but higher quality tracking device. As the PowerGlove uses this tracking system also, the PowerGlove's ultrasonic transducer and sensor systems were both used in the prototype glove device created. The flexion sensors from the original PowerGlove were also re-used in the prototype glove device. The forearm, transmitter, and flexion sensor electronics were modified to improve the data quality, make the glove more machine-independent, and give a much higher degree of control over the glove's components, which is valuable if the glove requires repairing. A hardware device consisting of gates, latches, counters and timers could be used to control the glove's functions, but it was decided that a general purpose microcontroller would offer a more flexible solution. Such microcontrollers are essentially general purpose CPUs to which various input and output enhancements have been added for a variety of methods of interfacing with other devices. The microcontroller chosen for this task was the Motorola 68HC11, as it featured many inbuilt hardware and software subsystems that made it ideally suited for the task. If a more general purpose CPU was used, these hardware and software systems would need to be added to give adequate functionality to control the glove, increasing the cost and complexity of the glove control system.

Modifications to the PowerGlove electronics were performed on all three of its main features: the position and orientation tracking system, the finger flexion sensors, and the serial communications section.

7.3 Motorola 68HC11 Microcontroller

The 68HC11 microcontroller is an integrated circuit designed to function as a stand-alone unit for controlling one or more devices, and to perform this task it is equipped with a number of hardware and software subsections that give it great flexibility for a large range of applications. These subsections cover serial (RS-232) communications, parallel input and output, analogue to digital conversion, complex timer operations, monitoring systems to check the microcontroller's operation, and other functions (see [Mot91] for an in-depth description of the 68HC11 and its capabilities).

Of these systems, only some are needed to perform the functions of a glove device controller where the glove device functions in a similar way to the original PowerGlove. The analogue to digital conversion section is capable of handling input from the finger flexion sensors; of the two different serial subsystems available only one is required for communication with the host machine; and the powerful timer and counter system is used for controlling the glove's ultrasonic transmitters and processing of input from the ultrasonic sensors.

7.3.1 Free Running Counter System

The 68HC11 has a 16 bit counter built into the chip that cycles from 0 to 65535 then repeats the cycle again from 0. This counter is activated immediately the chip is powered up and continues counting regardless of the other subsystems on the chip. It can only be changed by resetting the chip, causing the count to reset to 0 and begin its count cycle again. Other subsystems are connected to this counter in various ways, and it is this use of the counter that provides the main function of the position sensing system of the glove.

7.3.2 Input Capture and Output Compare Systems

The input capture (IC) and output compare (OC) systems make extensive use of the free running counter. For the IC system's operation, aspects of a electrical pulse applied to particular pins on one of the 68HC11's parallel input/output ports cause the current contents of the counter to be stored in a memory location. For example, a change from 0 volts to the positive supply voltage of the chip, applied to an IC pin, can cause the contents of the free running counter to be latched and stored in an internal data register when the change in voltage occurs. It can also, if desired, cause an interrupt condition in the 68HC11's CPU, allowing a specific piece of code to be executed when the voltage change on the IC pin occurs.

The OC system has a similar function, except in reverse. At a specified time, the voltage on one or more OC pins can be set, reset, toggled or in a special case, set to a defined value. This allows cyclic patterns to be applied from the OC pin(s) with very high accuracy. The OC system can also be set to cause an interrupt when a change in a pin occurs. This allows the programmer to be able to alter the waveform applied from the pin, in a controlled, accurate and repeatable way.

These three systems make up the basic functioning of the position tracking system in the modified PowerGlove. The OC system is set up via a software initialisation to activate a given transmitter at a particular time. The signal from the transmitter is picked up via a sensor and fed to an IC pin, causing it to grab the current counter value. By having specific timer periods for each transmitter, it can easily be determined from the IC timer value which transmitter it corresponds.

7.3.3 Analogue to Digital Conversion and Serial Communications

An 8-bit, 8 channel, analogue to digital conversion system is provided on the 68HC11 microcontroller. This system is used to measure finger flexion from the sensors fitted into the PowerGlove, and gives a much higher degree of accuracy than the original PowerGlove's 2 bits per finger. The extra bits available allow for more advanced processing of finger values in the host (see Section 6.3.1), as well as being more flexible in what can be determined from the finger flexion values.

The serial communications system follows the RS-232 standard, which is available on almost all modern computer systems. This gives the microcontroller user a great degree of machine independence in the choice of host machine to connect to the microcontroller. Such independence of the host machine is very useful in a science centre environment, as there is no guarantee that any particular type of machine will be used exclusively for the exhibit, and the machine itself may be changed frequently. The original PowerGlove's form of serial communications is a proprietary protocol designed to communicate with the NES and required the use of an adapting device to enable it to be machine independent. This device does not improve the quality of data output from the glove, but merely translates it into the RS-232 protocol.

7.4 The Modified PowerGlove

To prototype a new design for the glove device, a PowerGlove was used as the starting point. This approach retained many of the desired features of the PowerGlove that would be useful in a final product, such as its robustness and the pre-moulded shape and lycra glove structure. If no major refinements to the PowerGlove casing were required, its body could be used in the new version of the glove device. The PowerGlove, having being designed for the home entertainment market is structurally very robust, much more so than the other commercially available gloves, and this property is extremely important for devices in a science centre environment. The electronics system of the PowerGlove had to be changed however.

7.4.1 Hardware changes

The main change as mentioned previously was to replace the PowerGlove's controlling CPU unit with a 68HC11 microcontroller. Some other changes were necessary to facilitate the use of the 68HC11. It was decided that to reduce the weight of the forearm section on the modified glove, the 68HC11 system would be mounted in a separate unit, and connected to the glove section via a cable. This setup required the finger flexion sensors to be amplified to avoid loss of signal and electrical interference when sent down the connecting cable. Control signals to the transmitters are fed up this cable from the 68HC11 to the glove section. Also a clock source for driving the transmitters needed to be fed up the cable to the glove section. This clock source needed to be synchronised with the 68HC11's CPU clock for reasons that will be presented in Section 7.4.2.

7.4.2 Control software

There are four main sections in the software, for each of the four functions the system has to perform. The glove's transmitters must be switched on and off at known times, received pulses must be stored, finger flexion values must be measured, and commands from the host must be acted upon. The I/O subsystems on the 68HC11 are primarily interrupt/event driven, and control of the glove's transmitters and capturing of the received pulses are both handled by interrupt routines, while the other two functions are performed in a continuous loop.

This main loop samples the Analogue to Digital ports connected to finger flexion sensors in the glove section, stores these values in memory, checks for a command on the serial line, and processes it if a command has been received. These commands are:

- Send the current position 'distances' (these are values stored by the input capture system representing the free running counter's value when a pulse was received; the glove's position in three dimensions is calculated in the host machine);
- Send the current finger samples from the A/D conversion system;
- Send a particular character (used for detecting if the 68HC11 is connected to the host and operating correctly);
- Quit to ROM monitor (the 68HC11 family have a ROM program burnt into them to enable programming from another computer). Under normal operation the glove system does not use this command, but it is available to aid reprogramming of the glove controller code.

The 68HC11 ROM contains routines for serial port input and output but these were not used, and new ones needed to be written, because the provided routines converted characters to 7 bits by stripping the most significant bit, and there was also a large degree of redundancy in the code, with a number of layers of subroutines that can cause problems with stack overflow. Variables and temporary storage in the 68HC11's onboard

RAM must fit into less than 256 bytes on most versions of the CPU, and have to share this space with the program stack, so it is possible to overwrite these variables by nesting subroutines too deeply.

Continuous output of samples as they become available was not used because this method of output is incompatible with some host machines. Sun Microsystems workstations, for example, can have problems when a continuous stream of data is fed to a serial port. Synchronisation between the 68HC11 and host machine also needs to be resolved for a continuous transmission method. However a continuous system would bypass some of the issues encountered in a polled system such as a requirement for unique counter samples, which are discussed below.

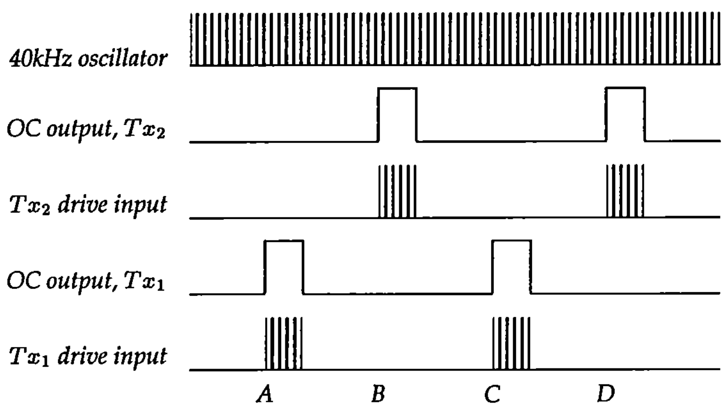


Figure 7.2: Glove transmitter control signals

Dealing with echoes is necessary to maintain correct operation of the system. Reception of an echo pulse will cause an input capture event, and if IC interrupts are enabled, an interrupt will be also be generated, causing the microcontroller to execute an IC interrupt routine. Each time the IC interrupt routines are executed, they disable the interrupt that causes them to be called, stopping themselves from being executed again, but they do not disable events, and the event flag can still be set by another IC event. In this situation, as soon as the IC interrupts are re-enabled an interrupt will occur, at a time it should not have. To cope with this the output compare system clears any pending IC events before starting each new transmitter cycle, and then enables the IC interrupts.

The output compare interrupt routines for controlling the glove's transmitters work in the following way. Under interrupt control the transmitter pins activate and deactivate the ultrasonic transmitters at specific times, corresponding to certain values of the free-running counter. The outputs of the two transmitter control pins are shown in Figure 7.2. The same routine is used for each stage in the waveform, so it must determine what stage in the waveform caused the current interrupt, because all of the transitions on the transmitter pins will cause the same routine to be executed. There is a lot of common code for each of the transmitter transitions, and only one interrupt is used for controlling them, so the code is shorter if a single routine is used, rather than one for each different transition.

Once the cause of the current interrupt has been determined, the output compare system is set up for the next stage in the waveform, because the OC system works by setting an OC pin or pins to a chosen value and then generating an interrupt, rather than generating an interrupt and having the interrupt code (or some normal operation code) perform some action on the OC pin. The latter method of operation is possible, but cannot guarantee accurate timing of control of the pins, whereas the prior system maintains very accurate timing control. The stages of operation are shown in the following cycle, repeated over and over in normal operation of the glove:

Start : The initial step, when the OC system is set up to turn on transmitter X at time A;

Time A : Set up OC system to switch off all transmitters at time B;

Time B : Set up to switch on transmitter Y at time C;

Time C : As for time A, but for time D;

Time D : As for start time.

Once the system determines what part of the waveform it is currently at, it can perform any actions that may need to be done at particular times. If the system has just passed times B or D, then all that is required is to set up the OC pins for the next point in the waveform. If it is at times A or D, then a number of other operations should be performed. The OC system needs to be set up to switch the transmitters off at the next point in the waveform; the input capture system needs to be reset to discard any echo effects (see above); a *wave number* variable is updated (see later for a description of this variable); and the previous IC counter values must be copied to a save area to avoid being overwritten by capturing the next pulse. Also, because IC values are time-multiplexed (two transmitters sending pulses at different times, received by three sensors, each going to a separate IC pin), the IC values must be redirected to the storage space appropriate for the transmitter that this captured pulse came from.

The *wave number* variable is used by the host to determine if a set of distance values sent from the 68HC11 are duplicates of the previous set of distance values. This is needed if the host can request positions faster than the OC/IC system is generating them. The prototype of the modified glove system generated an approximately 30Hz sample rate for both transmitters (i.e. to get the distance values for each of the transmitters the full 16 bit count is used) with the 16 bit counter running from 0-65535 at 2MHz. However because the glove's position in the host's input driver is calculated on the average of the two positions to reduce effects of noise, the position is effectively updated at approximately 60Hz (the sample rate for any transmitter, irrelevant of which one it is). If this rate was lower than the rate of position requests from the host (which in the prototype modified glove system is approximately 40Hz), then the possibility could arise of duplicated positions being passed to the gesture recognition code, a situation that would cause incorrect recognition of gestures as described in Section 6.3.3. The gesture recognition code assumes that all positions fed to it from the glove input driver are valid samples, with no duplication of previous samples, so if the 68HC11 were to report duplicated values then the results described in Section 6.3.3 would occur.

Because of the period of time involved in transmitting a group of distances down a serial link from the 68HC11 to the host, there is a possibility of these distances being overwritten by the OC interrupt routine during the transmission. This only has a serious effect when the counter value for that distance is near a byte boundary (i.e. multiple of 256). If the low byte of the counter's 16 bit captured value is near the middle of a byte (e.g. `0x80`), then there will be no noticeable change in the distance; the value will represent the correct distance. If the change in distance crosses a byte boundary then the transmitted distance will be incorrect, and the glove will appear to have 'jumped' either closer to or further away from the sensor. This causes noise in the glove's position values, which can disrupt other code that uses these positions.

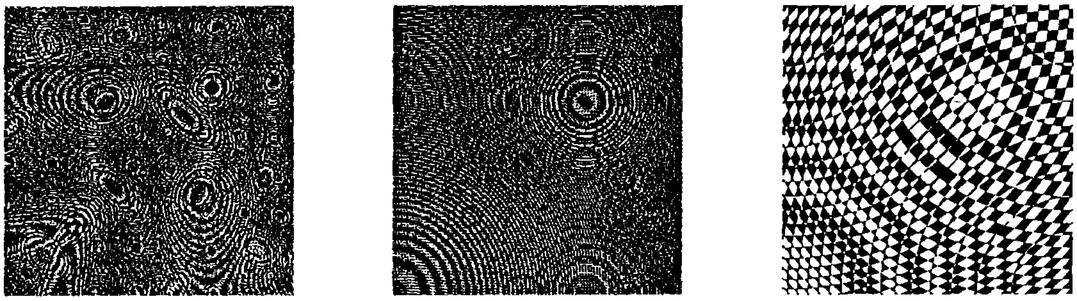
To avoid this a semaphore is used to ensure that the distances being transmitted won't be overwritten during the transmission process. When the distances are being transmitted, the semaphore is set, and the OC routine copies new IC counter values into a temporary storage space. The counter values are copied back to the normal distance storage area when the transmission is complete.

The input capture interrupt routines are comparably simpler than the output compare routine. Each IC routine (one is used for each IC pin) is triggered by a specific change in the voltage applied to its corresponding IC pin. When the routine is triggered, it copies the latched counter value to the appropriate storage space determined by the OC routine, and disables any further interrupts that would cause it to be executed again. This is to avoid the problem of echoes causing multiple counter values to be latched. The interrupts are re-enabled by the OC routine when it has just started a waveform (i.e. times A and C). This scheme ensures that only the first pulse received is recorded and all others are ignored. There is still a chance of echoes triggering the system if an echo arrives in the next pulse interval just after the OC routine has re-enabled IC interrupts, but the signal strength of the pulse in this situation is small enough for this situation not to occur. The echoes that are most likely to falsely trigger the IC routines are those with a path just slightly longer than direct line-of-sight, and these are ignored by the process described above.

If a transmitter is obscured from a sensor the IC system for that sensor will not be activated, and its interrupt routine will not be executed. In this situation the previous distance value is used instead, since there is no way of determining when the sensor will become unobscured again, and waiting for it to do so may hold up the reporting of samples. Because positions are averaged in the host system's input driver, the effects of this situation are reduced.

The range of the system is determined by how far a signal will travel from transmitter to the most distant sensor in the time between each transmitter's activation. This period of time is chosen to be the time that the free-running counter takes to count from 0 to 32678, or half of its full count range. At 2MHz which is the clock rate for the counter, this period of time is approximately 16 milliseconds, which corresponds to just over 5 metres from a transmitter to the most distant sensor, which is sufficient to cover the volume most users of the system are likely to move the glove in. Distances beyond 5 metres will not be correctly reported because pulses will be received in the next transmitter cycle, for the other transmitter. In practice this was not found to be a problem, as users did not operate the glove system beyond 5 metres from the sensors. Using the whole count range lowers the sample rate, and increases the range to 10 metres, which is far more than what is required for the system; using a quarter of the count range reduces the range to 2.5m, which does not sufficiently cover the volume users are likely to move the glove within, so this could not be used either. The calculations involved in three-dimensional triangulation are shown in Figure 3.6.

Another problem that increases with distance is resolution of the system. Each distinct measurable distance from a sensor can be visualised in three dimensions as a point surrounded by concentric spheres. Figure 7.3 shows cross sections taken at different distances away from the sensors. The granularity of the image is important; each solid area of colour represents a region in the cross section where resolved distance values are constant for each sensor. As the distance from the sensors increase, these constant-distance regions get larger, denoting a decrease in resolution.



(a) .5 metres (b) 5 metres (c) 50 metres

Figure 7.3: Resolution degradation at distinct distances from sensor array

As can be seen in Figure 7.3, positions further away from the sensors are resolved more coarsely than those close to the sensors. This has an effect on the accuracy of the system, making it less accurate at far distances than at close distances. Theoretical analysis of the prototype system at its maximum range of operation of 5 metres showed this effect not to be dominant at this range, and the resolving power of the system was not compromised. The theoretical resolution at this range is better than 1mm, but it varies across the cross-section, depending on the distance from the sensors, as can be seen from the patterns in Figure 7.3. Other effects such as lowering of the signal to noise ratio due to the distance between transmitter and sensor had a greater effect on the system than the resolving power effect described above. Background noise from fans, keys, and other ultrasonic sources tended to be sensed rather than the transmitters when the glove was distant from the sensors.

Each transmitter is expected to be turned on at a known time by the control code on the 68HC11. This requires a clock synchronised with the 68HC11's clock, that provides a signal to drive the transmitters. If an asynchronous clock is used, the transmitters will most probably not be activated at the time the control code expects them to be. This will effectively cause a delay in the transmitted ultrasonic signal, which will also be delayed in its reception. If the 68HC11 clock is running at an exact multiple of the clock used to drive the transmitters, then over a period of time the delay in the received ultrasonic signal will be constant. If the 68HC11 clock is not an exact multiple of the transmitter clock, then the delay in reception of the ultrasonic signal will drift, causing drifting or jittering of the calculated three-dimensional position in the host, which degrades the result.

These are just some of the issues that are important in designing a low cost glove device based on ultrasonic and strain gauge technology to operate in virtual reality and other applications. While the above mentioned issues are aimed primarily at a design based on the Nintendo PowerGlove, many of them are applicable to the design of any kind of glove device for use with a computer.

7.5 Summary

The Nintendo PowerGlove is a low-cost input device that is suitable for venturing into virtual reality research. However, for serious applications the PowerGlove itself is unsuitable, but provides a good starting point for a more versatile glove. Modifying the PowerGlove made it suitable for use in the "Virtual Maestro" science centre exhibit. This chapter has discussed one possible version of such a modified glove, describing its construction and functioning, and some of the factors that should be taken into account during the design phase of this and other low cost glove devices.

Chapter 8

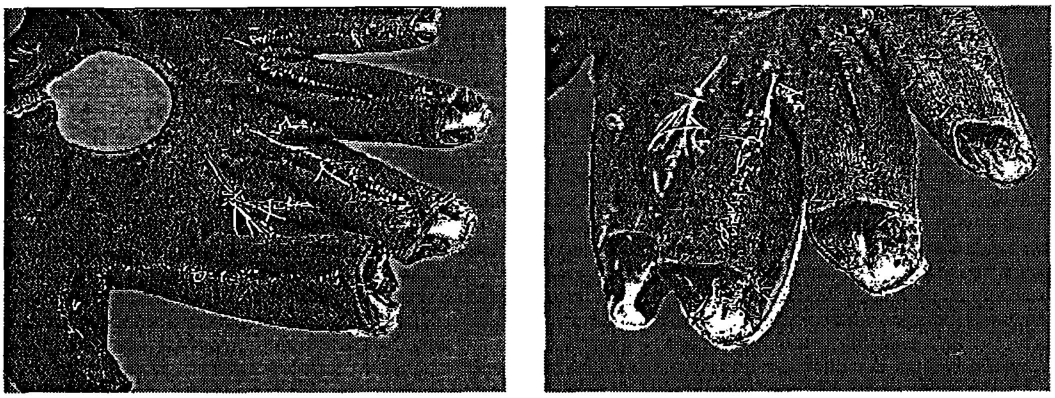
Evaluation of the system

Once the system was assembled it was installed at the *Science Alive!* science centre. Observations of the exhibit's functioning were taken over the first few days of operation, and on a number of occasions in the following months. The exhibit functioned satisfactorily, but a few problems were encountered.

8.1 Glove failure

Although the PowerGlove was designed for the rugged home video market, the environment of a science centre is extremely demanding, and the PowerGlove was not intended for this type of application. After its installation the system was operated for a number of months before being removed due to excessive and unforeseen wear on the glove devices. PowerGloves were used both in their original form and as a structural basis for the modified glove. They are constructed of a latex hand-shaped plate to which a lycra fabric glove is bonded (see Figure 7.1). The lycra glove section has a vinyl hand shaped plate woven to the back of it, which matches the latex plate in shape, and it is this vinyl section that is bonded to the latex plate. With the initial number of visitors to the science centre being in the order of 1000 per day, the glove device was being strapped and unstrapped on users' hands very often. As mentioned in Chapter 6.1, hand sizes between these users varied greatly with small hands of young children and large hands of adults being placed into the glove device. This had the effect of stretching the lycra in places to almost twice its original size.

This was not a major problem with the gloves because the lycra would shrink back to a smaller size after a period of inactivity. However, due to the large number of times the device was used by different people the lycra fabric started to tear in a number of places. One of the most noticeable areas where this tearing occurred, as well the stretching mentioned previously, was in the fingertips of the glove. Glove fingers that were affected in this way were less sensitive to flexion of the user's finger because the sensor is mounted between the vinyl on the glove section and the latex plate. Sensors on glove fingers that are stretched or torn register less flexion than undamaged glove fingers, and over time the glove becomes less sensitive to flexions in those fingers as the condition of the lycra deteriorates further (see Figure 8.1).



(a) (b)

Figure 8.1: Typical damage to the glove devices

A more serious problem with the construction of the PowerGlove is the way the sensors are mounted between the latex plate and the vinyl of the lycra glove section. Each sensor is mounted within a plastic jacket, to allow it to move freely when the user's finger flexes. Because there is no allowance for the vinyl part of the lycra glove section to move, stretch, or shrink relative to the latex plate at the area where they are bonded to each other, an inconsistency arises in the length of the two plates. This is caused by the plates being bent at different radii from the point of bending which in this case is a joint in the user's finger. Because of this inconsistency, the two plates pull together, sandwiching the sensor and its surrounding jacket together. This sandwiching effect is most prominent towards the fingertip, and the sensor is gripped here and pulled into its jacket. When this occurs connectors at the end of the sensor (for an external device to measure the sensor's resistance) are pulled against the sensor jacket, applying stress to them, and weakening the sensor and wire connections around this area.

When the user straightens their finger, the gripped sensor is then pushed in the reverse direction along the jacket. For reasons such as moisture in the jacket (causing the jacket to adhere to the sensor) and physical obstruction by the electronics mounting plate, the sensor can be impeded in its travel causing it to be compressed along its major axis. This puts a lot of stress on the sensor and usually causes a kink to appear in the plastic body of the sensor. The resistive ink coating at this point is often minutely cracked, breaking the circuit and making the sensor unusable.

The problems associated with the sensor failure are likely to occur in most mechanical sensor based glove devices because of the nature of mechanical sensors that require some form of motion to detect flexion. A design of an optical system with no moving parts is being considered as a possible solution to this problem.

With the high number of visitors to the science centre gloves wore out from sensor failure within approximately two to four weeks of operation. This was a very short period of operation for a glove, and the decreasing supply of PowerGloves made obtaining new ones difficult. A new design for the glove device based on a modular approach to its construction is planned for future development.

8.2 Aural quality

The quality of aural feedback from the synthesiser has a large effect on the perceived realism of the virtual instruments. Initially the system produced only simple tones from a square wave oscillator. This was adequate for testing the triggering and behaviour of the system, but did not create a sense of playing an instrument. The tones were fixed in amplitude, and had no varying timbral properties to indicate the type of instrument being played. When the instrument being played behaves, for example, as a piano does, and sounds very much like a piano, it is easy to suspend disbelief and assume that it is a piano. Suspension of disbelief is more difficult if the object either does not behave like a piano, or does not sound like one.

At the next stage of testing a frequency modulation (FM) based synthesiser was used for aural output. This improved the realism of the exhibit greatly, but still the timbral differences between sounds produced by this synthesiser and sounds from the real instruments were noticeable. The synthesiser was able to vary amplitude of played notes to convey a sensation, for example, of having struck a piano key with some force. However the timbral qualities offered by the FM technique of sound synthesis are sufficiently different to real instruments to have a marked effect on the realism of the exhibit. The FM synthesiser was capable of producing attack/decay/sustain/release (ADSR) envelopes that added to the realism of the system.

The current synthesiser in use is one based on pulse-code modulation (PCM) that stores digital samples of the instruments in its internal memory. This gives a very realistic sounding instrument, and enhanced the exhibit greatly. While all the instruments improved greatly in the change from an FM to PCM based synthesiser, the main noticeable instruments that benefited from this change in sound source were the 'piano' and the 'strings' instruments. While more complex synthesisers are available, a satisfactory result was achieved with the PCM system, and so it was retained in the exhibit. The use of a suitably powerful audio amplifier is also important for maintaining aural quality of the sounds produced. Low quality amplifiers can distort the sounds fed into them, and in this case such distortion would reduce the realism of the system.

8.3 Exhibit related aspects

How the exhibit was packaged had a number of effects on its use. The system was displayed in a large black box shaped like an upright piano, with the glove connected to the front of it. Many users on initially trying the system would not bother reading the instructions provided, but would put the glove on and expect the exhibit to perform some function. Typically, when nothing then happened, they would move up to the surface of the black box and tentatively poke at the region where a keyboard would normally be on a piano. This had a negative effect because the glove was out of the optimal operating area determined by the beam-width of the ultrasonic transducers used in the system (see the grey zones in Figure 8.2). Another aspect of the users' haste caused them to be unaware of the other available instruments. Typically they would only attempt piano playing gestures, and not explore other possibilities. Once shown the different instruments available, the users gained a much greater enjoyment and understanding of the exhibit.

The instructions provided by the science centre for operating the exhibit were not clear or informative. The purpose of exhibit instructions is to guide the user into the best method of using an exhibit, and to provide further background information for the user on the exhibit and its related topics. The instructions provided for the "Virtual Maestro" exhibit did not perform the function described above. Users were moderately confused about the exhibit and its function, as the reasonably abstract design for the exhibit's packaging did not suggest its function or method of operation.

Many users, and also visitors who did not use the system but watched others using it,

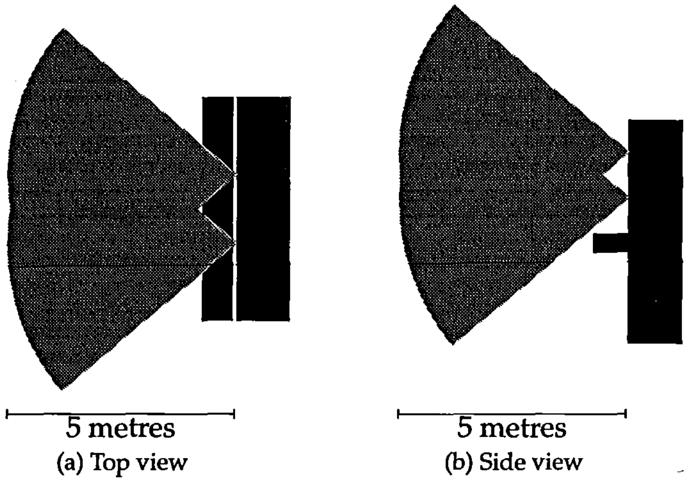


Figure 8.2: Operating volume of the glove device

wished to know how the system worked. If the instructions provided had been of a more informative nature, these people would have been able to learn about the system in a more controlled way. As the author was on hand at a number of occasions to explain the system and its operation, these users and visitors left knowing a little more about VR than they had done upon entering the science centre, but during other periods visitors and users would not gain a great deal of understanding about virtual reality. These visitors did not operate the system effectively because they were not aware of how it should be operated. Because of this they were often not aware of the possibility of repeatable cause-and-effect relationships between their actions and the sounds produced, apparently at random by the exhibit.

This makes the point that good instructions are required for an exhibit to be effective. The instructions for use should be unambiguous, and the explanation of how the system works should be in terms that many if not all visitors can understand. The instructions also need to be in a prominent place where any visitors likely to use the system can read how to operate it. Expectations and limitations should be stated in the instructions, to minimise confusion and reduce the necessity of demonstrator assistance. Users need to be informed that they should point the glove towards the black box, as the system will not function correctly or at all when the glove is directed elsewhere. They are not inherently aware of the behaviour and function of the system, and so the instructions should either suggest how to trigger all of the individual instruments, (e.g. "play the piano like this...") or at least a method for determining how to trigger them should be described (e.g. "move your hand and fingers like this").

8.4 Exhibit concept

The basic concept of the exhibit proved very popular with its users. The concept of being able to play virtual musical instruments was a novelty for them and this gave the exhibit a wide appeal. It was successful with children, who enjoyed the system immensely, especially the virtual drum kit. The 'rapid-fire' effect of the drums was therefore retained, and allowed younger users to gain a lot of enjoyment from the system, more so than probably would have been gained from using a traditional drum kit (i.e. one-shot) based system.

Skilled drummers noted that the layout of instruments in the virtual drum kit was not what they expected the drum kit to be, based on their experience with real drum kits. Pianists also noticed differences in the system, such as the large key size, and the behaviour of the fingers differing from what they were used to. Such differences appeared in the behaviour of chords, and the requirement to not flex the thumb. Most users did not notice the lack of hysteresis between notes on the virtual piano. Users inexperienced at the real-life version of the particular virtual instrument they were interacting with were less aware of the behavioural differences discussed above and discovered by more experienced musicians.

Because of the size of the keyboard and the behaviour of the fingers, it was difficult for users to play any melody more complex than "Mary had a little lamb" on the virtual piano; there was a reduction in chords available, the thumb could not be used, the little finger tied to ring finger gave only three usable fingers and only one hand was able to produce notes. Because of these limitations on the system a number of possible changes are being considered to enhance the apparent skill of the user. These changes are discussed in Chapter 9.

8.5 Other issues

One major aspect of the exhibit that was not addressed sufficiently was the issue of hygiene. The glove was used by a large number of people, all with varying skin conditions. Over a period of time the gloves became noticeably discoloured and gained a strong odour. Initially a modular glove device was planned, parts of which could be replaced, cleaned or repaired as required. This did not eventuate due to time and cost constraints, but is planned for the future version of the glove.

The system initially had a continuous sounding saxophone styled instrument at the start before the user activated the gesture based system. This instrument was to attract passers by when the system was not in use, and to show them the system was operational and not switched off. It also remained active until a user could activate some of the other available instruments. In theory, this continuous instrument was to be re-activated after a user had finished with the system, but the effect on surrounding exhibits of a continuous sound coming from this exhibit would become annoying over a period of time. One possible solution that was not tried because the exhibit was removed from the floor, is to observe the raw finger flexion values (i.e. before the normalisation process presented in Section 6.3.1), and determine when they were changing by more than a certain threshold level. This would limit the effects of noise falsely triggering the continuous instrument. If the flexion values have been below this threshold value for some length of time then the glove is presumed to be not in use, and the continuous instrument does not sound. When the flexion values go above this threshold, the continuous instrument would become active, until the user had performed some gesture to indicate that they wished to play one of the virtual instruments, at which point the normal instruments would become available. Observing the position values of the glove is not a satisfactory test, as these can be very inaccurate when the glove is out of its optimal operating volume, which is the case when it is sitting on the exhibit (see Figure 8.2).

8.6 Summary

The concept of the exhibit was successful as a teaching aid in conveying the principles of virtual reality to the general public. However, a number of points were of note in evaluation of the exhibit. The exhibit needs to have clear, concise instructions and information to explain to users the exhibit's functioning and purpose. It also needs to be constructed of as few moving parts as possible, to minimise stress on components and decrease maintenance periods. An ideal system would be un-encumbered, similar to the techniques used in Videoplace.

The exhibit's concept of playing virtual musical instruments is appealing to visitors to the centre, but it should probably be redesigned to give it a more abstract appearance rather than a resemblance of a piano form. This is to reduce bias towards playing a piano only, because of the packaging of the exhibit. Because of the lack of graphical feedback, and the overlap in gestures for some of the instruments, the intrigue of exploring the exhibit to discover what instruments it can mimic may have to be sacrificed to allow visitors to benefit from trying all the available virtual instruments. This will reduce the interest value of the exhibit, but will increase its effectiveness as a learning tool.

Chapter 9

Conclusions

This thesis has covered the design and development of a hardware and software computer system for demonstrating the concept of virtual reality to visitors at a science centre. The system has been developed for a reasonably low cost, well within the range of most science education centres, and the exhibit has proved to be effective as a learning tool for the chosen topic. Use of mass-produced consumer items as components in the system has contributed greatly to the system's simplicity and to its low cost. However, the level of robustness common on such components, while higher than that of laboratory equipment, is below what is required for a science centre environment. If the system were used as a demonstration tool which for example, was presented to schools and used to teach pupils, then the consumer components used would probably be sufficient.

By itself the exhibit is not sufficient to inform users adequately of a new and novel concept such as virtual reality. Information and instructions need to be supplied to complement the exhibit if it is to function autonomously as a teaching system.

A number of issues in the design of the exhibit have been recognised and dealt with to allow it to perform its function with available components. Some of these issues probably would not have arisen if other types of components had been used.

9.1 Future work

> 'Going to the Feelies this evening, Henry?' enquired the Assistant Predestinator. 'I hear the new one at the Alhambra is first-rate. There's a love scene on a bearskin rug; they say it's marvellous. Every hair of the bear reproduced. The most amazing tactual effects.'

- Aldous Huxley's *Brave New World*

A number of enhancements to the glove device and translator system are possible to improve the system. Some of these possible enhancements have been described earlier in this thesis, such as alteration to the model of piano operation, discussed in Section 6.3.7, and chord production when a number of fingers are flexed, discussed in Section 6.3.8. Addition of another glove for the user's left hand is another possible change to the system. This would allow instruments to be modelled more realistically, with gestures that more closely represented triggering of the instruments being simulated. This improves the realism of the system because multi-input instruments based on hand gestures and postures from both hands, such as violins, do not need to have their inputs mapped onto unused degrees of freedom of a single glove, thereby changing the nature of the instrument.

Tactile feedback in the glove device would greatly enhance the user's sense of interacting with a virtual instrument, because another sensory stimulus is agreeing with the aural sense, strengthening the user's mental model of something being present. If the visual sense were 'disabled' while using the system — by having the user operate the glove device from behind a screen so that the view of their hand is obscured from them, for example — then the conflict between their visual and aural senses would be reduced since they would be unaware of what they cannot see. If tactile feedback were also used, then there would be little conflicting information for them to assume they were using anything but a real instrument. Some tactile feedback systems are described in [AB92, MOyS+90, Rhe91].

To make the exhibit more enjoyable and interesting for novices, an auto-accompaniment system is a possibility that could be added. In this situation, the user would begin playing an instrument, and the computer would follow up with notes and sounds on other instruments. Such a system has been implemented by Dannenberg [Dan84], and by Vercoe [Ver84].

Packaging and presentation changes are possible future alterations to the system that may be undertaken. Addition of abstractly shaped physical objects may be used to give the user a better idea of where to make movements for activation of instruments. The glove device needs improvement or elimination, for example by using a baton.

A new exhibit is also planned, the "Instant Orchestra", based on work by Morita et al. [MHO91], where the computer responds to gestures made by a user holding a baton, and making the motions of an orchestra conductor. The system will have pre-stored files of musical scores, and the actions of the user and motions of the baton will determine parameters such as volume of various orchestra sections, and tempo of the piece while the computer 'plays' it through the synthesiser. Both the baton and the controlling software will be based on the Radio Baton and the Conductor program, both described in [Mat91]. Other features may be added to give the computer a more expressive, 'human orchestra' feel, such as the expert system approach suggested by Johnson [Joh91].

9.2 Summary

A low cost system has been developed for the teaching of virtual reality concepts to the general public. This system is well within the finance range of most science centres, and its novelty makes it a useful learning tool.

Acknowledgements

I would like to thank the following people in the creation of this manuscript: Drs. Tim Bell and Gill Bryant for kindly loaning equipment used in the construction of the Virtual Musical Instrument Exhibit; *Science Alive!* for the opportunity to undertake this thesis; Mike Cusdins of the Electrical Engineering Department, Canterbury University for assistance in redesigning the glove device; a cast of thousands (well, at least 5!) who helped with ideas, thoughts and other input; those two furry felines which saved me buying an alarm clock (but cost a fortune in cat food); and a very special thanks to my fiance, Margaret Gerberi for her support throughout and beyond the final stages, Tinker Bell for her lively character, Magic Carpet for his fun and games, and Walter Elias Disney, who kept me happy.

Bibliography

[AB92] Steve Austakalnis and David Blatner. *Silicon Mirage: The Art and Science of Virtual Reality*. Peachpit Press, 1992.

[App91] Apple Computer Inc. *MIDI Management Tools Set v2.01*, October 1991.

[App92] Hugh L. Applewhite. Position tracking in virtual reality. *CyberEdge Journal Monograph* (Supplement to Issue 8), 928 Green Hill Rd.-Mill Valey, CA 94941, USA, March 1992.

[AS84] M. B. Alt and K. M. Shaw. Characteristics of ideal museum exhibits. *British Journal of Psychology*, 75:25–36, 1984.

[Bal92] Robert D. Ballard. The 'Jason' project: hi-tech exploration promotes students' interest in science. *Technological Horizons in Education (Journal)*, pages 70–74, 1992.

[BCW90] Timothy C. Bell, John G. Cleary, and Ian H. Witten. *Text Compression*. Prentice Hall, 1990.

[Bel92] Tim Bell. Computer science for the uninterested: Designing displays for a science centre. *Computers in New Zealand Schools*, 4(1):40–46, March 1992.

[BF92] Will Bauer and Bruce Foss. GAMS: An integrated media control system. *Computer Music Journal*, 16(1):19–24, 1992.

[Bio92] Frank Biocca. Will simulator sickness slow down the diffusion of virtual environment technology? *Presence*, 1(3):334–343, 1992.

[BM91] Francis Balaguer and Angelo Mangili. Virtual environments. In D.Thalmann and N.Magnenat-Thalmann, editors, *New Trends In Animation and Visualization*, pages 91–105. John Wiley, Lausanne, 1991.

[BS74] Robert P. Burton and Ivan E. Sutherland. Twinkle Box - a three-dimensional computer input device. In *National Computer Conference*, pages 513–520. AFIPS Press, 1974.

[Cha90] Xavier Chabot. Gesture interfaces and a software toolkit for performance with electronics. *Computer Music Journal*, 14(2):15–27, 1990.

[Chr92] Michael Christel. Virtual reality today on a PC. *Instruction Delivery Systems*, 6:6–9, July 1992.

[CLF84] C. Cadoz, A. Luciani, and J. Florens. Responsive input devices and sound synthesis by simulation of instrumental mechanisms: The Cordis system. *Computer Music Journal*, 8(3):60–73, 1984.

[CLF90] Claude Cadoz, Leszek Lisowski, and Jean-Loup Florens. A modular feedback keyboard design. *Computer Music Journal*, 14(2):47–51, 1990.

[CMJ91] Opcode systems new music software for Apple Macintosh. *Computer Music Journal*, 15(1):80–82, 1991.

[Dan84] R.B. Dannenberg. An on-line algorithm for real-time accompaniment. In *Proceedings of the International Music Conference*, pages 193–198, San Francisco, 1984. Computer Music Association.

[Deu92] Diana Deutsch. Paradoxes of musical pitch. *Scientific American*, pages 70–75, August 1992.

[DL92] Paul DiZio and James R. Lackner. Spatial orientation, adaptation, and motion sickness in real and virtual environments. *Presence*, 1(3):319–328, 1992.

[Ebe92] Sheldon M. Ebenholtz. Motion sickness and oculomotor systems in virtual environments. *Presence*, 1(3):302–305, 1992.

[Egl90] Howard Eglowstein. Reach out and touch your data. *BYTE*, pages 283–290, July 1990.

[FDJ77] Henry Fuchs, Joe Duran, and Brian Johnson. A system for automatic aquisition of three-dimensional data. In *National Computer Conference*, pages 49–53. AFIPS Press, 1977.

[Fen93] David Fenner. The pladar tracking system. *VR News*, 2(5):8–10, June 1993.

[Fis90] Scott S. Fisher. Virtual interface environments. In Brenda Laurel and S. Joy Mountford, editors, *The Art of Human-Computer Interface Design*, pages 423–438. Addison-Wesley, 1990.

[Fis91] Scott S. Fisher. Virtual environments: Personal simulations and telepresence. In Sandra K. Helsel and Judith Paris Roth, editors, *Virtual Reality : Theory, Practice and Promise*, pages 101–110. Meckler, 1991.

[FL92] Gary Ferrington and Kenneth Loge. Virtual Reality: A New Learning Environment. *The Computing Teacher*, 19:16–19, April 1992.

[Fol87] James D. Foley. Interfaces for advanced computing. *Scientific American*, 257(4):82–90, 1987.

[Fri91] Mark Fritz. The world of virtual reality. *Training*, 28:45–50, 91.

[FSP92] Martin Friedmann, Thad Starner, and Alex Pentland. Synchronization in virtual realities. *Presence*, 1(1):139–144, 1992.

[FvFH90] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. IBM Systems Programming Series. Addison–Wesley, 2nd edition, 1990.

[Gar92] Ann Garrison. Digital improvisation. *MacWorld*, page 131, July 1992.

[Gib93] W. Wyatt Gibbs. Body English. *Scientific American*, pages 94–96, August 1993.

[Gra92] Joeseph D. Gradecki. PCVR Magazine, volume 1 issue 1. Gradecki Publishing, 1706 Sherman Hill Road # A, Laramie, WY 82070, 1992. Software provided with this issue of PCVR Magazine for interfacing to the Nintendo PowerGlove.

[Har91a] John Hart. In search of a definition of virtual reality. Master's thesis, Arts, Victoria University of Wellington, New Zealand, 1991.

[Har91b] John Hart. Through the Looking-Glass With Alice, Kitty and a Few Mates: Virtual Reality at Capitol Discovery Place, Wellington's Children's Museum. In Geoff Wyvill, editor, *The 12th New Zealand Computer Conference*, pages 267–274. New Zealand Computer Society, 1991.

[Har93] John Hart. VR alive and well - what about NZ? The Dominion Infotech Weekly, April 1993. *New Zealand newspaper, April 19 issue, pp 13.*

[HB86] Donald Hearn and M. Pauline Baker. *Computer Graphics.* Prentice-Hall International, 1986.

[HD92] Richard M. Held and Nathaniel I. Durlach. Telepresence. *Presence,* 1(1):109–112, 1992.

[Hee92] Carrie Heeter. Being there: the subjective experience of presence. *Presence,* 1(2):262–271, 1992.

[HR92] J. Hettinger, Lawrence and Gary E. Riccio. Visually induced motion sickness in virtual environments. *Presence,* 1(3):306–310, 1992.

[Joh91] Margaret Johnson. A computer music system that follows a human conductor. *IEEE Computer,* pages 30–34, July 1991.

[KL90] R. Benjamin Knapp and Hugh S. Lusted. A bioelectric controller for computer music applications. *Computer Music Journal,* 14(1):42–47, 1990.

[Kru83] Myron W. Krueger. *Artificial Reality.* Addison–Wesley, 1983.

[Kru91] Myron W. Krueger. *Artificial Reality.* Addison–Wesley, 2nd edition, 1991.

[LKL+92] Norman E. Lane, Robert S. Kennedy, Michael G. Lilienthal, Kevin S. Berbaum, and Lawrence J. Hettinger. Profile analysis of simulator sickness symptoms: Application to virtual environment systems. *Presence,* 1(3):295–301, 1992.

[MAB92] Kenneth Meyer, Hugh L. Applewhite, and Frank A. Biocca. A survey of position trackers. *Presence,* 1(2):173–200, 1992.

[Mat91] Max V. Mathews. The Radio Baton and Conductor Program, or: Pitch, the most important and least expressive part of music. *Computer Music Journal,* 15(4):37–46, 1991.

[Mer90] Mark L. Merickel. The Creative Technologies Project: Will training in 2D/3D graphics enhance kids' cognitive skills? *Technological Horizons in Education (Journal),* 1990.

[MHO91] Hideyuki Morita, Shuji Hashimoto, and Sadamu Ohteru. A computer music system that follows a human conductor. *IEEE Computer*, pages 44–53, July 1991.

[Mot91] Motorola. *HC11 Technical Data Manual, MC68HC11E9*, 1991.

[MOyS+90] Margaret Minsky, Ming Ouh-young, Oliver Steele, Frederick P. Brooks Jr., and Max Behensky. Feeling and seeing: Issues in force display. *ACM Computer Graphics*, 24(2):235–243, March 1990.

[MS92] Michael E. McCauley and Thomas J. Sharkey. Cybersickness: Perception of self-motion in virtual environments. *Presence*, 1(3):311–318, 1992.

[New92] Rocking all over a virtual world. *New Scientist*, 136(1852/1853):20, December 1992. 19/26 issue.

[Pan93] Veronica S. Pantelidis. Virtual reality in the classroom. *Educational Technology*, 33:23–27, April 1993.

[Pau91] Randy Pausch. Virtual reality on $5 a day. In Scott P. Robertson, Gary M. Olson, and Judith S. Olson, editors, *Reaching Through Technology*, pages 265–270. ACM SIGCHI, Addison-Wesley, 1991.

[PCC92] Randy Pausch, Thomas Crea, and Mathew Conway. A literature survey for virtual environments: Military flight simulator visual systems and simulator sickness. *Presence*, 1(3):344–363, 1992.

[Pen85] Bruce W. Pennycook. Computer–music interfaces: A survey. *ACM Computing Surveys*, 17(2):267–289, June 1985.

[Poo94] Bob Pool. Virtual volleyball. Los Angeles Times, February 1994. *Los Angeles newspaper, February 5th issue, pp B1.*

[PT92] Ken Pimentel and Kevin Teixeira. *Virtual Realtiy: Through the new looking glass*. Intel/Windcrest/McGraw-Hill, 1992.

[Ral94] Stuart Ralston. Virtual reality science centre exhibits. Master's thesis, University of Canterbury, New Zealand, 1994.

[Rhe] Thomas Rhea. Clara Rockmore: The Art of the Theremin (Music review). *Computer Music Journal*, 13(1):61–63.

[Rhe91] Howard Rheingold. *Virtual Reality*. Summit Books, 1991.

[RM90] Dean Rubine and Paul McAvinney. Programmable finger-tracking instrument controllers. *Computer Music Journal*, 14(1):26–41, 1990.

[Rub91] Dean Rubine. Specifying gestures by example. *Computer Graphics*, 25(4):329–337, July 1991.

[Rum90] Francis Rumsey. *MIDI Systems and Control*. Focal Press, 1990.

[She92] Thomas B. Sheridan. Musings on telepresence and virtual presence. *Presence*, 1(1):120–126, 1992.

[Shn87] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, 2 edition, 1987.

[ST91] Rory Stuart and John C. Thomas. The implications of education in cyberspace. *Multimedia Review*, 2:17–27, 1991.

[Stu91] David J. Sturman. *Whole-hand Input*. PhD thesis, Architecture and Planning, Massachusetts Institute of Technology, 1991.

[Sut65] Ivan E. Sutherland. The ultimate display. In *Proceedings of the IFIP Congress*, volume 2, pages 757–764, November 1965.

[Sut68] Ivan E. Sutherland. A head-mounted three-dimensional display. In *1968 Fall Joint Computer Conference*, volume 33 of *AFIPS Conference Proceedings*, pages 757–764, 1968.

[Tuf86] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Box 430, Cheshire, Conneticut 06410, December 1986.

[Tyr89] Terian Tyre. Live broadcasts from the ocean floor bring new depth to science education. *Technological Horizons in Education (Journal)*, pages 42–46, 1989.

[Utt89] R. Uttal, William. Teleoperators. *Scientific American*, pages 74–79, December 1989.

[Ver84] B. Vercoe. The synthetic performer in the context of live performance. In *Proceedings of the International Music Conference*, pages 199–200, San Francisco, 1984. Computer Music Association.

[Vir92] Virtual Research Flight Helmet. Product Information, Virtual Research, 1313 Socorro Avenue, Sunnyvale, California, USA, 1992.

[VRN92] The MIT holographic video display. *VR News*, 1(6):12–15, July 1992.

[WRB+90] Jih-fang Wang, Azuma Ronald, Gary Bishop, Vernon Chi, John Eyles, and Henry Fuchs. Tracking a head-mounted display in a room-sized environment with head-mounted cameras. *Proceedings of the SPIE - The International Society for Optical Engineering*, 1290:47–57, 1990.

[Xte92] Products for Virtual Reality – Tactools Tactile Feedback for Virtual Reality. Product Information, Xtensory Inc., 140 Sunridge Dr., Scotts Valley, California, USA, 1992.

[Yam93] Philip Yam. Surreal science. *Scientific American*, 266:103–104, February 1993.

[ZLB+87] Thomas G. Zimmerman, Jaron Lanier, Chuck Blanchard, Steve Bryson, and Young Harvill. A hand gesture interface device. In John M. Carroll and Peter P. Tanner, editors, *Human Factors in Computing Systems and Graphics Interface - IV*, pages 189–192. ACM SIGCHI, ACM Press, 1987.