Mathgrasp: The design of an algebra manipulation tool for visually disabled mathematicians using spatial-sound and manual gestures

Philip A. Harling, Robert Stevens and Alistair Edwards
Human-Computer Interaction Group
Department of Computer Science
University of York, YORK, YO1 5DD
United Kingdom

Tel: +44-1904-432765 Fax: +44-1904-432767
E-mail: {philiph, robert, alistair}@cs.york.ac.uk

16th October 1995

Abstract

This paper reports on the design of Mathgrasp, an algebra manipulation tool that combines spatial sound and manual gestures to provide a non-visual application for visually disabled school children and students. The design seeks to ensure that a listener has the same speed and accuracy of control during manipulation as a sighted colleague.

We describe the pitfalls of a speech-based presentation and how they can be mitigated using spatial-sound and manual gestures. These two modes are combined appropriately, considering their inherent limitations, in an attempt to provide a direct manipulation style interface–a paradigm that is usually considered to be purely visual.

1 Introduction

Direct manipulation is a well known paradigm for interaction and most current mouse-based interfaces are considered to be exemplars of the type. The concept is taken a step further with virtual reality interfaces in which the user's hands can interact with three-dimensional graphical objects - through instrumented gloves or other devices. This paper describes the design of an interface which uses such gestural interactions, but to manipulate objects which are rendered in sound rather than as graphics. The interface arises from the peculiar constraints of the application which is to provide a means of enabling blind people to manipulate mathematical representations.

The representations in question are text-based ones, mainly algebra. Auditory representations of such material have been developed in the Mathtalk Project (Stevens and Edwards 1994; Stevens and Edwards 1993). Those representations have be developed and combined with tactile (braille) representations in the Maths Project (Edwards and Stevens 1994). However, it is suggested that whereas the interaction in the Maths Workstation is driven by keyboard and speech input, gesture-based input would truly be more direct and hence easier to use and more powerful.

The principles of Mathtalk's design were the compensation for lack of external memory and poor control of information flow which characterise the listening reader. We describe how the external memory provided by print on paper and the control afforded by the visual system have many of the elements of a direct manipulation user interface which a visually-disabled student currently lacks in any system for using algebra notation. The basic idea of Mathgrasp is to use the speech representation developed under Mathtalk and Maths presented in three-dimensional space in front of the user and then to allow the user to interact with the sounds directly through manual gestures.

Mathtalk has transformed the passive listener into an active reader. However, when Mathtalk is extended to the reading, writing and manipulation of algebra it would benefit from the development of a direct manipulation style of interaction. This would be accomplished using spatial sound and gestures to implement reading and manipulation operations.

The discipline of mathematics relies upon paper as other disciplines rely on the spoken word. Whenever two mathematicians meet or a lone mathematician works, paper is used to write down algebra, graphs, diagrams etc. The print on the paper acts as an external memory. It relieves the reader from the burden of retaining information that is often dense, complex and large. In speech, remembering the gist of an utterance is often enough but with algebra notation the lack of redundancy means that the loss of a single item can completely change the meaning of an expression. The form of the print on the page also helps the reader parse the expression and derive the relationships between the objects on the page (Kirshner 1989; Larkin 1989). For a sighted reader, the typography of algebra also allows the selection of parts of the expression appropriate to the current task. The external memory combined with the speed and accuracy of the visual system that give control over information flow make reading an active process in which the comprehension of the information takes priority over the retrieval of the information.

The listener lacks both external memory and control over the information flow. The speech signal is serial and transient and once it has been heard is gone. The auditory system can select information from the current environment via the attentional system, but it cannot select sounds that are in the past or in the future in the way that eyes can select words from a page. Even with recorded speech the control over information flow is slow and inaccurate. This lack of control makes the listening reader passive, rather than active. When listening to aurally presented text concentration is often lost, complex information is hard to process and retain, and review of information is tedious.

Mathtalk sought to make the passive listener an active reader. Spoken presentation of algebra was given some of the qualities of external memory. Prosodic cues were used to make the speech more memorable; introduce parsing cues into the utterance and reduce the mental workload associated with the listening task. A structure based browsing style was used to give fast and accurate control over the information flow to give active reading. This is discussed in section 2.

Despite the improvements in the presentation, Mathtalk still does not *display* an expression. The sound comes from a single source, in this case a loud-speaker, and it lacks the spatial components of printed external memory. It is now possible to ad a spatial component to sounds produced by a computer (Wenzel, Wightman, and Kistler 1991). These sounds can be made to appear 'outside the head', such that they may be pointed at, rather than simply appearing between the ears, as is the case with stereo. The addition of location cues gives redundancy by allowing the listener to use these cues instead of purely relying upon structural information.

1.1 Direct manipulation

Redundancy is not the only benefit that the additional of spatial sound provides. Location information also makes possible the use of a direct manipulation interface. The key elements of a standard direct manipulation interface are, (Shneiderman 1982; Shneiderman 1983):

1. continuous representation,
2. the use of physical actions (rather than complex syntax) that allow
3. incremental, easily reversible operations whose effects are immediately 'visible.'

The first criterion is obviously met by the expression display as it is displayed in front of the user with the left-most term of the expression to the left of the user and the right-most to the right. The other two criteria are met by the use of manual gestural interaction.

Actual physical manual gestures are introduced into the interface so that the user is able to directly manipulate terms and elements of the expression. By using an instrumented jacket and gloves (Figure 1.1), the position of the arms and hands relative to the user can be measured and these measurements are then used as input to a gesture recogniser. The recogniser decodes hand shape and hand location and uses this information to issue commands to the expression browser.

`datawear.eps`

Figure 1: Instrumented suit and gloves used in the Mathgrasp program.

The paradigm that Mathgrasp uses is to simply allow the user to grab objects and rearrange them using intuitive, deictic gestures as if they were real-world objects. As the expression is now rendered using spatial sound, the user is now aware that the $3x$ of the equation $3x + 4 = 7$ has its own location, to the left of the user, and similarly, 7 is to the right. Hence the the levels of psychological *engagement* and *directness* (Hutchins, Hollan, and Norman 1986) in Mathgrasp are high, thus presenting an interface that presents a good match between the way the user thinks about the problem and how Mathgrasp represents it. This thereby reduces the mental effort of the user in determining what needs to be done and how to do it.

There are a few limitations on the use of the gestures that imposes constraints on when and what type of gestures can be used. For example, the onset of fatigue due to reaching for objects in mid-air can cause the interface to be unusable over long periods of time. Therefore the gesture set has been designed so that there is a simple mapping between the height of the rendered object and the co-ordinates of the position of the hand so the objects appear to be resting on the desk.

The use of *appropriate* gestures is also an important issue. Gestures have only been used where they make sense, such as for selecting objects, moving them using grasping gestures and throwing them away. The problem of selecting gestures that have no natural symbolic meaning for commands such as **calculate**, has been avoided by using a more appropriate mode, such as speech or keyboard. Extending the direct manipulation paradigm to perform tasks that it is not wholly suited to has been recognised as a problem (Frohlich 1993) in general, not just one attributable to gestural interaction. The design of the gesture set and the selection of appropriate gestures is discussed later in this paper in section 4.

1.2 Example

Throughout this paper, we will use the example of the solution of the expression, $3x + 4 = 7$ to highlight features of the design for Mathgrasp. It also demonstrates the power of the combination of the printed expression and the visual system and how complex even a simple algebraic task can be. This should make clear the difficulty of performing mathematical tasks outside the visual modality, and to clarify the job that Mathgrasp is required to do.

The following example shows all the possible steps that could be done by a very naive mathematician.

1. $3x + 4 = 7$ (write expression)
2. $3x = 7 - 4$ (rearrange +4)
3. $3x = 3$ (calculate)
4. $x = \frac{3}{3}$ (divide by 3)
5. $x = 1$ (simplify fraction)

The paper holds all the moves that have been performed, releasing the user from the task of remembering the current status so that she is able to concentrate on the next stage, including calculations and goal-seeking. In the task, the mathematician performs rearrangements of the expression, calculations and reviewing of the information. Mathgrasp needs to emulate these activities in an easy, non-intrusive manner.

2 The Mathtalk Program

The Mathtalk program was developed to enable blind students and school children to read algebra notation using speech and non-speech sounds (Stevens and Edwards 1994; Stevens and Edwards 1993). The principle aim was to transform a blind user from a passive listener to an active reader. To effect this change, the twin themes of the design were to increase the efficacy of a auditory presentation as an external memory and to give the listening reader control over information flow. These were achieved with the use of prosodic cues in the speech output and a browsing language to move around an expression's structure.

As well as simply acting as a memory aid, the form of printed algebra also helps in mathematical tasks. The main function of algebra notation is to show the grouping of symbols and the relationship between those symbols. The order of precedence is instantiated within the typography of algebra and these cues aid the reader in parsing the expression (Kirshner 1989). For example, the spacing around operators divides an expression into chunks that correlate with the major parsing points. In addition, such typographic cues aid the sighted reader in moving around an expression and selecting portions to read (Haber and Haber 1981).

2.1 Adding Prosody to Spoken Algebra

The types of cue that appear in print are typically missing from a synthetic speech presentation of algebra. Not only can the explicit cues be missing or ambiguously presented, but the implicit spacing cues that aid parsing are also missing. Mathtalk uses the prosody of natural language to reintroduce some of these cues. Prosody is the rhythm and intonation of a language and can be thought of as the nonverbal information content of speech (Edwards 1991). Prosody can signal the mood and intention of the speaker, as well as improve memorability of spoken material and the structure of that information.

Mathtalk takes advantage of the two latter points to improve the usability of the spoken presentation (Stevens and Edwards 1994; Stevens, Wright, and Edwards 1994) In speech some, but not all, pauses and pitch changes are associated with clause boundaries (Crystal 1975; Garnham 1989). For example, the lexically identical utterances:

- The last time we met Robert, was horrible.
- The last time we met, Robert was horrible.

In oral reading the correlation between prosodic cue and syntactic juncture increases to 80% from the 60% seen in conversation (Garnham 1989). Listeners are able to recover the structure of natural and algebraic utterances using such cues (Streeter 1978; O'Malley, Kloker, and Dara-Abrams 1973).

When synthesising spoken natural language it is often difficult to derive the structure to which these prosodic cues are associated. This is not the case with algebra notation; the purpose of the notation is to indicate grouping, so all the necessary information is present. Earlier researchers have proposed some rules for algebraic prosody and shown that listeners can use these cues to insert structure into expressions (Streeter 1978; O'Malley, Kloker, and Dara-Abrams 1973). Part of the research undertaken during the Mathtalk project was to develop and extend these rules. In brief, these cues include:

- Pauses placed before operators such as $+$ and $-$, but only outside complex structures such as fractions and parentheses.
- A pause is placed on the side of a relational operator next to the longer side of an expression.

- Only initial terms of an operand and a relational operator are emphasised.
- Pitch declines over the length of an expression from an initial pitch defined by the length of an expression.
- Local pitch changes occur about a term, with an initial rise for the onset of a term and a fall during the term.
- Pitch falls for the onset of complex objects; speed is increased, no pauses or emphasis occurs and pitch rises at the end of a complex structure.

These rules were implemented within the Mathtalk program[1]. The evaluation of the effects of prosody on the presentation of algebra showed that significantly more structure was recovered used prosodic cues; more of the content (letters, numbers and other symbols) was retained and mental workload was significantly reduced (Stevens, Wright, and Edwards 1994). The addition of prosodic cues made the synthetic speech presentation more usable by reintroducing some of the features of an external memory into the presentation. The chunking and rhythm of the prosodic presentation made the expression more memorable. Removal of surplus lexical cues, used to delimit boundaries, reduced length leading to greater retention of the contents.

Whilst the addition of prosody improved the presentation, it still lacked many of the qualities of a display. There are cues that indicate groupings and the memorability is improved, but it lacks the spatial component common to displays. Spatial information adds redundancy to the information presented. A sub-expression is delimited by parentheses and the size and location of that object can be given in spatial as well as structural terms of reference. Adding a spatial component to the Mathtalk program will be the subject of the next section.

2.2 Browsing Algebra Notation

The second theme of the design for Mathtalk was the addition of control over information flow to make the reading active, rather than a passive listening. A simple structure based command language was used to give this control (Stevens and Edwards 1994). A series of commands were based on a set of **action** and **target** words. Actions include **current**, **previous** and **next**. Targets include **expression**, **term** and **item**. A command is made by combining an action with a target. The full set of actions and commands can be seen in Table 1. A simple mnemonic mapping of initial letters from the words to the keyboard is used.

One feature of the Mathtalk program led to finer control over information flow than that provided with browsing alone. This was the hiding of complex objects. A complex object is one in which more than one algebraic term is grouped together spatially or by some parsing mark. During browsing such objects are not spoken in full, but referred to only by their structural type. The expression $3(x+4) = 7$ would then be rendered as '3 times a quantity equals seven', when the **current level** command is invoked. Similarly the expression

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

could be reduced to 'x equals a fraction'. This hiding means that large objects do

[1] The best way of understanding how these rules are implemented is to see a demonstration of Mathtalk. A mock-up is with detailed explanation is available on the World Wide Web at the URL *http://dcpu1.cs.york.ac.uk:6666/hci/aig/robert/mathtalk.html.*

Action	**Target**
Current	Expression
Next	Term
Previous	Item
Show	Quantity
Into	Fraction
Out-of	Numerator
Begin	Denominator
End	Superscript
From	Relational
Upto	

Table 1: **Action** and **target** words used to form browsing commands in the Mathtalk program. An action is combined with a target to form a valid command.

not have to be spoken as one large chunk; strictly delimited boundaries of constructs and provides overviews of expression structure to the listening reader.

It can readily be seen that the commands in Table 1 fall naturally into a spoken form and cover most moves a reader may want to make when moving around an expression. This command language proved easy to learn by users, who made few errors and extended their own command set by combining words into commands they had not been taught. They could move rapidly to any portion of the expression, without too much disruption of the reading process. The basic moves provided by the language were combined to give larger tactics and strategies.

The command language used to mediate the browsing within Mathtalk was well designed. It demonstrated that giving the listening reader control over information flow transformed that reader into the active partner in the interaction. Despite this demonstration of the principle there are some potential problems in the use of such a command system.

Whilst the set of actions is probably complete, the set of target structures could expand greatly. This would mean that the tactic of using initial letters would no longer be viable. This would be likely to make the learning of the system more difficult.

The command system is relatively simple, but only provides for the reading of an algebra expression. Once the system is extended to writing and manipulation of algebra, designing an integrated keyboard-based language would prove difficult. One option, being explored in the Maths project, is to use speech recognition as an input system for the command language and capitalise on its natural spoken form. The Maths workstation will also allow manipulation. What has been spoken as output will form the current selection. Simple **move** and **copy** keyboard or speech input commands will be used to perform basic manipulations.

Perhaps the fundamental problem with the command language design is the lack of an element of direct manipulation. For example, with a command language, manipulating the expression $3x+4=7$ so that $+4$ is on the right-hand side involves selecting the four, moving to the end of the expression and requesting that the selection is relocated. In contrast, a direct manipulation approach would be to pick up the plus four and then simply place it in the desired location. The benefit of direct manipulating over a command language is that the same object is used for both input and output, giving a coherent, seamless interaction rather than a series of disjointed commands. By combining a spatial display with gesture-based input we can develop an auditory direct manipulation tool to perform algebraic tasks.

3 Spatial Sound in Mathgrasp

Within Mathgrasp the expression is split into its component parts and then displayed using synthetic speech from left to right in front of the user, using prosodic cues to make the structure of the expression clear. The number and spacing of these parts is limited by the user's ability to discriminate them and the capabilities of the spatial sound hardware.

As well as hearing sounds the human auditory system can tell where in the environment those sounds emanated. Sounds produced by computers are at best stereo, but often come from a single loudspeaker. So when an expression is spoken, all the information comes from only one source. On the printed page, an expression has spatial information as well as structural information. This means that information can be processed in both structural and spatial terms, thus adding a level of redundancy to the reading interaction. A person reading and manipulating algebra in the auditory mode could also take advantage of such information. When reading an expression, apparent movement in space would confirm the browsing moves the reader had made. A spatial component could make 'moving an object over to the other side of an expression' more efficient and satisfying or more natural.

The technology is now available to simulate the spatialisation of sound. The capabilities of such systems will be described, before proposing a design for the inclusion of a spatial component into the Mathtalk program. The design described in this section is also being implemented as part of the Maths workstation.

It is thought that there are two principle ways that humans perceive the location of sound sources. The first is the *duplex* theory (Lord Rayleigh in Moore 1989). Pitt and Edwards (Pitt and Edwards 1991) use the binaural stereo successfully to to provide directional and positional information to find a target on an auditory display with reasonable speed and accuracy. However, Wenzel (Wenzel, Foster, F.L.Wightman, and D.J.Kistler 1989) points out that when using headphones, this gives sound that appears to be between the ears, rather than outside the head. The external ears (pinnæ) shape the spectra of the sounds in a manner that is highly dependent upon direction (Shaw 1974) and localisation is degraded when this shaping is removed.

This natural process can be simulated by using head-related transfer functions (HRTFs). These use the duplex theory, together with the effects of the pinnæby recording sound from two microphones placed within the ear canals. The resulting HRTFs can then be simulated how a sound will appear to a listener given coordinates in space. Devices such as the Convolvotron (Foster, Wenzel, and Taylor 1991) a high-performance system and the Beachtron, a cut-price version, provide the hardware to generate spatial audio.

Humans are very good at localising sound in front of them. In the median plane, movement can be detected over as little as one degree in a semi-ellipsis of $\pm 40^\circ$ azimuth and $+80^\circ$ elevation (Strybel, Manligas, and Perrott 1992). As the sound source moves around to 90° to the vertical plane through the nose (opposite the ear) movement has to occur over 40° to be detected. Due to the orientation of the ears, front-back localisation is much worse than localisation in front of the listener. Fortunately this is natural for the display of a paper based writing system such as algebra notation. In Mathgrasp the emphasis will be placed on movement in this direction.

Some evaluation has been performed on localisation performance with some of the commercial spatialisation products. Crispien and Fellbaum studied the ability to point at and select an object display in space using the Beachtron card manufactured by Crystal River Engineering. They found that, with this absolute localisation only a grid of five objects horizontally and three vertically, could be reliably discriminated.

With such a method, Mathgrasp could display very little of an expression at any one time. In addition, the accuracy of such pointing might cause frustration during reading. With this method, the user would have to remember the location of any one object, or all objects would be displayed concurrently, in order to locate any one object. The design described below avoids any need to have to find the absolute location of any object, making the interaction flow more easily.

A simple design would be used for the spatial component in Mathgrasp. In Mathgrasp, greater use would be made of relative movement than pointing to absolute position of sound. Similarly, greater use of the spatialisation from left-to-right than up-and-down movement would be made. An audio screen would be defined in front of the user, in an arc covering 80° from left-to-right. Such an area would give the best discrimination of sound movement. How much could be fitted onto this screen would depend on the technology used and the distance the expression was displayed in front of the listener. This distance would depend on how comfortable the listener was with a sound 'in their face' or too far away.

The simplest use of the spatial component would be to notice movement rather than absolute position. As the listener traverses an expression, each spoken object would be placed at a new x-coordinate sufficiently distant from the prior object such that the movement is noticeable. With the current technology this distance is likely to be much less than that required for pointing to absolute location. Movements in output as a result of browsing moves would help in confirmation of movement and that the correct or intended movement had been made. This would be particularly useful during manipulation.

Some difficulties are likely to be encountered in mapping prosodic cues into a spatial display. There is a mismatch between prosodic cue and spatial cue in the presentation of superscripts. The term ax^2 is all spoken on a falling tone, where the mapping from print would be to place the exponent above the x. As discrimination is less good in the vertical direction and the prosodic cues bind the superscript closely to the term, vertical display would not be used.

Similarly, a fraction is vertically juxtaposed, where the temporal juxtaposition in speech might dictate horizontal alignment of numerator and denominator spatially. In this case, a regression after completion of the left-to-right rendering of the numerator, combined with a lowering in space for the onset of the denominator, could prove useful in reinforcing the structure of the fraction.

As will be seen below, the formatting of the expression within the display space is determined by where the user starts the reading interaction. Similarly, when hidden objects are entered, the initial point of the new environment is determined by the placement of the gesture.

4 Using Gestures at the Human-Computer Interface

Imagine you are trying to explain a diagrammatic proof of Pythagorus' Theorem to a colleague over the telephone. The proof will require you to describe a simple diagram of triangles and squares. You will say things like, "There's a triangle in the bottom left, a large one ... now put a square adjoining the triangle's hypotenuse... yes, that's right, that's the top-most line, now mark this line x," and so on. During your explanation you will naturally want to attempt to either draw the figure with a pen and pencil, or mark out the diagram in the air or on a desk using gestures. Unfortunately, your colleague is unable to see any of your gestures and will have to be content with your verbal explanation.

The telephone forces you to use only speech to explain your diagram by removing all other communication channels, especially the visual channels such as the gestural and facial channels. When one mode of communication is more suited to an explanation than another the removal of the most suited mode can be extremely frustrating, as this example demonstrates. Another common example that elaborates this point is the description of people's faces using only speech; faces are difficult to verbally describe and I would rather show a photograph or sketch of someone's face than having to try and explain face shape and how someone's eyebrows meet, etc. So, we can clearly see that our preferred choice of communication mode is strongly influenced by the applicability of that chosen mode to the current task.

Gestures play a natural part in normal everyday speech. They provide clues about how large the fish is when we say, "It was *this* big," and hold our hands several feet apart, or what object we are talking about when we say, "Give me *that* and I'll wrap it up for you," and point to the box of chocolates in the corner of the room. The urge to use gestures to communicate is present with us from birth; indeed, it has been noted (Trevarthen 1986) that long before an infant is able to use adult language, she is able to manipulate objects and gesticulate to her parents to communicate her wants and needs.

So, we can see that manual gestures have a variety of uses. They can clarify what is spoken, they can intuitively be used to describe situations (locations and sizes of objects in space and their relative motion, etc.) that would be difficult to describe using other communication modes and with less effort. People use gestures from birth and are therefore very skilled in their use and gestures also form a fundamental part of deaf sign languages.

Clearly, gestural communication is an integral and important part of everyday human communication; however, the use of manual gestures at the human-computer interface is nonexistent. Indeed, the lack of attention of interface designers to exploit the natural skilled abilities of humans at the interface has long been known as Buxton (Buxton 1986) notes on the use of, in this case not hands but feet, at the interface – *"The feet are totally ignored ... [yet most users] have well-developed skills in their use."* Therefore, it seems important and appropriate to investigate the use of gestures at the interface.

4.1 Gesture input technology

Gestural interaction is a novel input mode and requires suitable technology to measure the user's gestures. Measurements about the hands, their location, finger position and movement can be collected in two ways, either actively or passively. Data can be collected passively using a video camera and actively use an instrumented suit, or three-space tracker if only hand location is required. The technologies have natural advantages and disadvantages that are related to how intrusive they are.

Instrumented suits allow measurements to be taken on the joints that are of interest, they require no complex processing to determine where the hands are, and the measurements are always available even when fingers and hands occlude one another.

Conversely, video cameras have the benefit that the user is not required to dress up in a suit that is laborious to don and is physically connected to the computer, restricting the users movements. However, cameras do have the disadvantage that they are susceptible to changes in the environment, such as lighting, background, shadows etc., and that important joints can occlude one another.

Three-space trackers come somewhere between gloves and cameras; they can detect the location of the users hand in space with reasonable accuracy (reference) as long as there are not large local electromagnetic fields nearby (unlike with the use of be and in a modern office environment). However, they only require the user to hold the tracker and not dress up in a suit, but they are unable to determine hand shape unless combined with a glove or camera, and so also inherit the problems associated with them.

Naturally, the choice of input technology depends upon the task in hand. Applications that require precise movements to be measured are more suited to using glove technology rather than video cameras, whereas applications that are to be used in public where dressing up in a suit is not practical, or require grosser, more general movements are more suited to a video camera.

4.2 Algorithms

One of the problems with the use of gestures at the interface is that gestures must be recognised and interpreted, in a similar way that speech must too. This is a difficult task as the production of gestures not only vary when they are made by different users, but also vary when they are made by the same user. Hence techniques and algorithms have been developed in order to recognise them, for example see (Baudel and Beaudouin-Lafon 1993; Fels and Hinton 1995).

Gestures can be split into two gross categories that aid recognition, namely *postures* and *gestures.* Postures are hand shapes that are static and involve no dynamic motion of shape of the hand or its position in space. Gestures, conversely, are those that involve the gamut of finger and hand motions. Therefore it is not unusual to expect the recognition of postures to be much easier than that of gestures.

Posture recognition can be reasonable accomplished using numerous techniques including edge maps (Uras and Verri 1995) to recognise 25 postures with an accuracy of 99% and artificial neural networks (Harling 1993) to recognise five postures with 96% accuracy. Posture recognisers usually only handle postures that are well separated and face problems when they are expected to recognise postures that occur sequentially, and this presents a barrier towards the recognition of natural, fluidly connected gestures. This barrier is known as the *segmentation problem.*

There are two problems that occur frequently to prevent the use of naturally made gestures. First the segmentation problem is a difficult problem to solve and many recognisers have failed at this point because of a lack of a structured approach. A related second problem is that many recognisers have limited use as their posture set is chosen for *ease* of recognition rather than *applicability.* This causes recognisers to fail when they are then applied to recognise genuine application gestures, thereby causing the user great frustration with low accuracy. This is a problem that we wish to overcome in Mathgrasp.

When a gesture set is not designed for ease of applicability but rather for the ease of recognition the user is forced to undergo long intensive training. This was done with Glove-Talk (Fels and Hinton 1993) (and Glove-TalkII (Fels and Hinton 1995)). In this case a set of unnatural gestures are used that can be recognised well, but are then inapplicable and cannot be generalised to other situations.

Both of these problems (segmentation and the use of unnatural gestures) can be solved with the introduction of a structured approach that focuses on the problem of the recognition of *general* gestures, rather than specific ones.

Posture recognition. A successful posture recogniser has been developed and implemented using artificial neural networks (Harling 1993). The posture recogniser was able to successfully distinguish five hand shapes of the fingerspelling of American Sign Language (ASL) with an accuracy of 96% for unseen users. This work was done before the current data suit (TCAS Datawear) was available, and used a Mattel Power Glove. The Power Glove has a very low resolution of finger bend and improvements in recognition beyond the reported 96% are thought to be due to the limited resolution of the glove. The higher resolution of the TCAS Datawear is expected to solve this.

Segmentation. The segmentation of *natural* gestures is very important for gesture recognisers if the use of gestures at the interface is to be practical. Emphasis is placed upon the segmentation of the user's natural gesture, rather than requiring that the user make gestures in a prescribed way to aid automatic recognition as it is entirely possible to ask the user to put a pause between each atomic gesture. This however, is extremely inconvenient and impractical for the user and is akin in speech recognition to putting a pause between each word and enunciating each word clearly. Thus the study of the coarticulation effects between gestures is extremely important for the production of a natural recogniser. Algorithms have been developed (Harling and Edwards 1995) to segment simple postures and are to be used in the Mathgrasp gesture recogniser.

4.3 Problems with Gestural Interaction

The use of gestural interaction is desirable as its benefits include amongst others, naturalness, intuitiveness, the exploitation of preacquired motor skills and for certain applications and situations it can be the most suited input modality. To make full use of of suitable gestural interaction, various issues that arise from the inherent nature of gesture communication have to be considered when designing an interface. Some of these issues stem from the current input technology and associated recognition algorithms, whilst others emerge from the general nature of gesture and their appropriate use.

It is generally recognised that there are three major practical issues with gesture-based input: fatigue from poorly designed gestures, physical mobility constraints due to the input technology and the immersion syndrome.

Fatigue. If the user is constantly requirely to keep their hands raised in the air for an extended period of time, or if they are expected to make many rapid motions then we can expect them to fatigue quickly. This has long been known with touchscreens (Sears 1991; Ahlstrom, Lenman, and Marmolin 1992) and Baudel and Beaudouin-Lafon 1993 have also found this with their *CHARADE* system (Baudel and Beaudouin-Lafon 1993).

Physical constraints. Due to the nature of the input transducers, either camera or glove, there are physical restraints that are placed on the user. With a camera, the user must always ensure that she is within the camera's view and does not go beyond the frame. With a data suit, the user is physically attached to the computer with the signal wires and is unable to move far from her desk. She must also put on and remove the suit every time that she needs to interact with Mathgrasp. Data suits also become uncomfortable, tiring and hot to wear for extended periods of time.

Immersion syndrome. As with other types of modes, for example speech, the *immersion syndrome* is also present with gestural interaction (Baudel and Beaudouin-Lafon 1993). The *immersion syndrome* is when the user is forced to interact with the computer because her moves are always being monitored. For example, in Mathgrasp a 'grabbing fist' is the command to select an object. Without a solution to the immersion syndrome in Mathgrasp, everytime the user went to grab her cup of tea to drink from, Mathgrasp would interpret this grabbing action as the **select** command.

Apart from constraints to do purely with the input technology, there are also constraints placed on what gestures may be recognised with the current state of recognition algorithms and the appropriate choice of gestures. As we have seen in the previous section, the development of gesture recognisers is hindered with problems that have yet to be solved, such as the segmentation problem. In order to produce a gesture set that is able to be recognised well and accurately with current recognisers the following points should be considered.

1. The postures should be simple and not involve both hand motion *and* finger motion, as this is difficult to recognise well.

2. The postures should be as different as possible in shape from each other in order to aid discrimination, but not at the expense of choosing unnatural gestures.

4.4 Appropriate gestures

The whole point of this section on gesture input and technology has been leading up to the use of appropriate gestures at the user interface. However, we have not yet given any criteria to determine what are appropriate gestures. Sturman discusses the appropriateness of the use of hand gestures in his thesis (Sturman 1992, pp.76-77) and lists as the requirements for the selection of a natural gesture: as pre-acquired sensorimotor skills, an already existing hand sign, and that the task control maps well to the hand action.

It is obvious that gestures that will work well as a command as those that already have a meaning in the physical world, although those that are symbolic are difficult to use as they have different cultural meanings. Deictic gestures, that is those for pointing at objects and manipulating them, are good to use in a command language as they already have a strong everyday meaning that everyone who moves objects in their physical environment will be aware of. The only danger point here is that the interface that uses them should use them in a manner that is consistent with the physical world.

Finally, there are the physical aspects of gestures that have to be considered, two of them we have already discussed: immersion and fatigue. Another consideration for the design of a gesture set is that whatever gesture is used, the interface must be able to cope with the gesture being made with *either* hand. This not only allows left- and right-handed people to use the same interface, but also allows them to exchange hands should they become tired. As in the real-world, tasks that are difficult to perform with one hand can made easier to do with the use of another hand. The use of both hands should also be considered when designing a set of appropriate gestures to give an interface that is natural to use.

5 Transforming Mathtalk to Mathgrasp

This section discusses the transformation of Mathtalk into Mathgrasp by the application of spatial sounds and gestural interaction. The general manipulation model is presented first, then five examples are given that show how interaction takes place, highlighting how objects are read, selected, marked, moved, deleted, how hidden objects are dealt with and how expressions that are larger than the current expression space are manipulated. It should be pointed out that the manual gestures in Mathgrasp are not used to write expressions. There are no natural gestures for writing, so their inclusion in the Mathgrasp program would be contrived and awkward. In addition, within the gesture set we use, some commands need to be given via another mode. For example, there are no gestures for 'calculate' or 'copy' and these commands given with speech or keyboard will be fused with the manual gesture for selecting and moving text, as described below.

5.1 The Manipulation Model

Mathgrasp regards the working space as a three-dimensional notepad. The user works upon an expression in a central portion of this pad and the margins of this working space can be used to hold notes and jottings. This is in effect a buffer that is equivalent to the cut-and-paste buffer in a text-editor.

The expression is displayed in front of the user as if it were upon a vertical 'audio screen'. Gestures are made upon the desk in front of the user as if he or she were pointing at something on a page (Figure 5.1 shows a mock-up of Mathgrasp being used to select an item $3x$.) These horizontal movements are mapped to the vertical screen display. This is analogous to the way that mouse movements are mapped onto the pointer location displayed on a visual display unit. This mapping is applied to reduce fatigue experienced by the user and also to solve the problem presented by the immersion syndrome. By always requiring the user to manipulate objects on the desk in front of her, pwithe.eps

Figure 2: The use of pointing in Mathgrasp: the left hand is used as a marker while the right hand is used to select part of the expression, $3x$. Note that the expression is rendered vertically whilst the user makes gestures on the desktop to combat *fatigue* and the *immersion syndrome.*

gwithe.eps

Figure 3: A grabbing hand is used to move selected objects within the expression space.

she is able to use any of the Mathgrasp command gestures (table 2) outside of this space and they will not be interpreted.

The space in which the expression is displayed, the *expression space* is constrained by several factors

- physical constraints of the user,
- physical constraints of the desk and
- the resolution of the audio screen provided by the spatial sound.

These constraints on the expression space form a window in which all or part of the expression can be placed. This has consequences for the design of the gesture set and manipulation of larger expressions that will emerge in the examples below.

The basis of the gesture set is to use both hands: one to point at the expression and manipulate it, and the other to simply mark places within it. The *reading finger* points at objects in the expression with a typical pointing gesture, using the index finger extended from a clenched fist on the dominant hand. A similar gesture on the non-dominant hand is used to remember object positions when selecting more than one object. The reading finger is also used to perform large movements within the expression space. It makes gestures to grab objects (see Figure 5.1), put objects, throw objects away, to open and close hidden objects, and to scroll the expression within the expression space. Grabbed objects can be either put in the expression, held in the hand, or placed in the margins of the page for later use.

Up to this point we have have only discussed moving through an expression serially. The benefit of spatial sound is that the location of an object can be directly determined. To go directly to any object in the expression space, the user can simply point at the appropriate position on the desk. This avoids the potentially tedious task of listening to all objects in order to reach a particular goal.

5.2 Basic Example

In this example we demonstrate the basic gestures for moving around an expression, selecting, deleting and moving objects. We also show how speech could be used to issue non-deictic commands that do not have a natural iconic gesture.

1. **State** $3x + 4 = 7$

   Goals Read the whole expression.

   Actions The reading is initiated by immersing the reading finger into the expression space. An open-handed sweep from left to right over the expression area invokes a full utterance.

   This step shows how the *immersion syndrome* can be avoided. Gestures are only recognised when they happen at the notepad, as is exemplified by placing the reading finger into the expression space to define where the beginning of the expression is placed. Gestures can be made outside of this space, for example, nose scratching, typing, writing, drinking tea, waving friends, or even command gestures, and yet they will not be acted upon. This is similar to turning towards people to speak to them, that is, it initiates dialogue.

   Placing the reading finger into the expression space has important ramifications. The location of the finger defines where the start of the expression will be fixed in the expression space. Whenever a gesture is made that involves the beginning of the expression, either initiation or return, then the location is fixed again even if it was previously fixed at a different place. The spatial rendering of the expression is measured relative to this point. This is done because the user has a limited range of movement as well as other physical limitations previously discussed, for example desk space, etc. It also avoids the user having to find the beginning of the expression when absolute reference might be inaccurate, and that the extremes of an expression can always be referenced quickly and accurately as they are important.

The gesture to read the expression is a sweeping expansive gesture from left to right over the expression that suggests, 'all of this.' Note that although this gesture has a deictic basis, it is more of a iconic command. While the hand does not necessarily sweep over all the expression, it is the entire expression that is read, not just the portion swept over. A command that is absolutely necessary in a speech based interface is the command to mute the speech output. An obvious gesture for this is a chopping movement onto the desktop to halt the speech flow.

State $3x + 4 = 7$

Goals Select the term $+4$.

Actions The hand moves back to the beginning of the expression in a vertical arc over the expression. The reading finger moves over the expression to the $+$. This is marked with the retaining finger, then the reading finger moves over to the 4 and the reading hand grabs and retains the selection in a clenched fist.

Moving the hand in an arc over the expression does two things. The first is to move the reading finger back to the beginning of the expression, but without reading the expression backwards. This is accomplished by removing the finger from the expression space in a vertical arc and then reimmersing it at the beginning of the expression. The reimmersion of the finger acts as a 'beginning' command as in the previous step.

In order to select +4 the user needs to find the location of the +. To do this, she needs to carefully read the expression and so uses a different gesture. Instead of the expansive sweeping gesture to read the entire expression, the reading hand forms a pointing posture suggesting selection of an individual object, rather than many objects. The finger is then moved along the expression from left to right with the finger on the desktop (see Figure 5.1, and the terms are read out as the finger passes over them at the user's rate of motion. Keeping the finger on the desktop avoids the *fatigue problem*.

This is similar to the motion used when children are learning to read and place their finger under the word they are currently reading and thereby the design uses previously learnt skills and connotations.

Once the user has found + its location needs to be retained in order to select the whole of +4. This is achieved by marking its position with a static pointing posture using the non-dominant hand. The selection can been completed with the reading finger behaving as before. Note that either hand can be used as the marking hand to allow for people for people with different handedness. A right handed person will typically use their right hand (and hence right index finger) as the reading finger, and their left hand to mark selections. However, left-handed people will use their left hand to read with and their right hand to mark. Therefore, the design of Mathgrasp allows either hand to be used as either the reading or marking hand.

The ability to swap reading hands overcomes two basic physical problems. The first is that large expressions can be read without the user straining across large distances, for example trying to read the rightmost end of an expression on the very right of their body with their left hand. The second problem occurs when the hands are placed such that the reading and marking hands must cross-over, for example if the left (reading) hand is reading an expression that is on the right of the user, then the right (marking) hand is used to mark it, and the left hand continues reading to the left. Crossing the hands over is extremely awkward and clumsily and being able to use either hand as the marking or reading hand solves this problem. However, an extremely left-handed (or right-handed) person who wishes to continues using their dominant hand as the reading hand can still continue to do so.

Both these problems can always be avoided by the user if necessary. As the expression can be reorientated at any point, it can always be placed at the most comfortable position to be read by using the 'refixing' gesture. The design of Mathgrasp is flexible enough to allow the user to use whichever method is more suitable and comfortable at the time.

Once the selection has been made, $+4$ can simply be removed from the expression by using a natural grabbing gesture. This is a pure deictic gesture that has a direct mapping to objects in the real world–once grabbed the selection remains in the fist (see Figure 5.1. We believe that because grabbing is a very basic and inherent human emotion, the haptic feedback that the transformation that the hand makes from a pointing posture to a clenched fist during the grabbing gesture provides extremely strong and superb feedback about the state of the selection.

The selection of an object places Mathgrasp into a selection mode whereby commands behave differently. It has always been observed that strong haptic feedback should be related to interface modes (Sellen, Kurtenbach, and Buxton 1992). In contrast, if a copy of the selection needs to be made, no natural gesture for 'copying' exists. Therefore it would seem more reasonable to use a speech command, 'copy,' spoken while while grabbing the selection as the hand moves out of the expression space. This shows the power of the fusion of speech and gesture. The result of this action is that the copied selection appears in the fist of the reading hand.

2. **State** $3x = 7$

   Goals Append the $+4$ that is retained in the fist, to the end of the expression.

   Actions While retaining the grabbed object in the fist, the fist moves in an arc to the end of the expression (with the same movement as the *beginning* command, but in the opposite direction); then opening the fist releases the sequestered object to the end of the expression, after 7.

   The movement to the end of the expression is the counterpart of that to the beginning. This gesture fixes the end of the expression where the gesture immerses itself in the expression space. Note that the end of expression movement is made with a clenched fist in this case, but the same movement can be used with other hand shapes, for example reading, to move to the end of the expression.

   The opening of the fist clenched about the selected object release it to be appended to the end of the expression. Again haptic feedback is used to reinforce the fact that the Mathgrasp is no longer in a 'place-state' and that normal manipulation gestures can be used.

3. **State** $3x = 7 - 4$

   Goals Replace the $7 - 4$ with 3.

   Actions The reading finger moves from the minus sign to 7, this point is then marked with the index finger of the left hand. The reading hand makes the end of expression gesture and then the selection $7 - 4$ is grabbed with the reading hand. While $7 - 4$ is in the buffer, the calculate command is issued. The content of the fist is replaced with the result, 3 and then the fist is opened and the contents are released to the point marked by the left hand.

The right-hand side can be selected as previously described, marking the location with the left hand. While the $7-4$ is held in the fist, the calculate command is issued using the spoken phrase, 'calculate.' Again this is an example of a situation where no natural gesture exists. The result, 3 replaces the contents of the fist with suitable audio feedback and then the fist is released and 3 is placed at the point marked by the marking hand.

Note that the fist is used in a similar way to a buffer in a traditional text editor, and that the gesture is similar to the way in which objects are manipulated and operated on in the physical world.

4. **State** $3x = 3$

   Goals Remove the initial 3 (of $3x$) and replace the final 3 with 1.

   Actions The hand is moved to the beginning with the beginning command, then the left hand marks 3 and the right hand grabs it, and throws it over the shoulder. The right hand makes an end of expression gesture, and while the index finger points at 3 it is replaced with a 1.

   All the operations have been discussed before, apart from the delete gesture. The gesture is based upon the everyday action of throwing unwanted objects away over the shoulder. This is taking material out of the expression space and dispensing with it.

   Note than when the 3 is replaced by 1, gestures are *not* used to perform the rewriting, instead something input mode more appropriate is to be used, for example speech or the keyboard. Similarly, the action of transforming the right hand side to the fraction $\frac{3}{3}$ cannot adequately be performed using gestures.

5. **State** $x = 1$

5.3 Absolute Pointing Example

The following is an example to show the benefits of spatial sound as part of the selection mechanism. We can also using pointing at the spatialised sound to perform moves and selections. An alternative to moving through each item to reach the object to be selected, is to point directly at that object. As the expression $3x + 4$ is spoken, it is spread out left to right in the expression space in front of the listener. Selection by pointing can take place by one of two means: either the user points directly at the point in space to select a spoken as it render from left to right, or after the rendering has finished the listener points at where he or she remembers the desired object is placed.

This type of pointing frees the listener from always having to start reading from the beginning of the expression. Such pointing allows random access to the expression. Given the limits of the technology, such direct pointing will not be reliably accurate, but may give quicker access and would, in time, become accurate.

5.4 Example: Scrolling

In some situations the expression to be rendered will not fit into the expression space. When the right-most boundary is reached, the user can then *pull* this point of the expression back to a comfortable point within the expression space that allows more of the expression to be read. In the following examples the box drawn around the expression or part of the expression indicates the scope of the expression space.

State $\boxed{5x^5 + 4x^4 + 3x^3} + 2x^x + x = 0$

Goal The reader wishes to read the expression, from left to right, until the end is reached.

Actions The user finds the beginning of the expression, the reading finger moves to the right, causing the contents of the expression to be spoken, until the user cannot comfortably reach any further right or the user is constrained by physical space. The user then holds on to the expression and pulls the held point back within a comfortable space. Then reading continues by moving the reading finger towards the right.

State $5x^5 + 4x^4 + \boxed{3x^3 + 2x^x + x} = 0$

5.5 Example: Reading Hidden Objects

As described in the section on Mathtalk, complex objects can be rendered by simply referring to their type. For example, speaking $3(x + 4) = 7$ as 'three times a quantity equals seven.' This helps the listening reader in controlling information flow. Mathgrasp needs a set of gestures for controlling movement in and out of such hidden objects.

1. **State** $3\boxed{\text{hidden}} = 7$

   Goal Display the contents of the hidden object.

   Description The user points at the hidden object, and then taps it twice to open it. The hidden object then expands to reveal its hidden contents (in this case, $x + 4$). The reading finger now points at the first item within the hidden object.

   The hidden object is selected as described previously. The sub-expression is simply rendered as 'a quantity'. The reading finger that points at the object taps twice on that object to suggest 'look at this'. This gesture opens the hidden object and the reading finger then points at the first object in the sub-expression, in this case x. The sub-expression now becomes the only information within the expression space. This constrains the reader to move only within the sub-expression and reinforces the structure of the whole expression.

2. **State** $\boxed{\text{x+4}}$

   Goals To close the expanded hidden object and return to the previous level.

   Description The user makes a squeezing motion anywhere in the expression space.

   State $3\boxed{\text{hidden}} = 7$

   The pointing gesture of the reading finger is transformed to an open hand, that then closes to bunch the fingers together into a 'squeezing' action. This suggests closing the previously opened complex object, hiding it once again. Like the other gestures used in the Mathgrasp interface this has a direct real-world counterpart. After closure, the new selection is the hidden object and the expression is formatted around that point.

5.6 Summary

Table 2 presents a summary of the gestural commands used in Mathgrasp along with an explanation of the gestures that are used to issue them (other, non-gesture commands such as **calculate** are not included.)

This set of gestures uses hand-shapes and actions that are easy to make. They have either a strong deictic form (such as reading or moving to the beginning or end of an expression) that is based heavily on a real-world physical model, or are strongly mimetic, such as **delete**, where the user mimes throwing the object away. As the gestures have a strong correlation with real-world activities, the gesture set should be easy to remember and use. Commands that break this correlation, such as **calculate** would be better issued in another mode such as speech – hence each mode within the design is used when appropriate. We propose that when gestures or keyboard input and corresponding output are heavily contrived, their use becomes less compelling.

It is also feasible to build a gesture recogniser that is able to easily recognise all of the command gestures. The gestures fall into two sets: Those that have static hand posture and hand motion,nd those that only have finger motion. As we have seen, it is easy to construct recognisers for both these types of gesture.

6 Conclusions

We have presented a simple model for a gesture based interface for manipulating algebra notation. It displayed the expression on a vertical audio screen using spatial sound and used a set of appropriate gestures to read and manipulate the expression. The design of the gesture set had several issues to address: the fatigue problem; the immersion syndrome; appropriateness of the gestures to the task and the ability of current recognition algorithms to successfully detect them.

The gestures were made at a horizontal desktop to solve both the fatigue problem and the immersion syndrome. Use of the desktop avoided the user having to hold his or her arms up for a prolonged time and consequently becoming tired. In addition, moving hands on and off the desktop enabled Mathgrasp to detect when *intentional* gestures were being made and thereby avoid the immersion syndrome. The relative nature of the placement of the expression's origin in the expression space avoided the reader having to search for the beginning and end of an expression giving more control to the reader.

An emphasis has been placed on the appropriateness of the gesture set. Gestures were only used if they had a real world counterpart; such as grabbing, pointing, moving and throwing. The attractiveness of the use of gestures was not diminished if there was not a satisfactory equivalent real-world gesture for a command and a more suitable mode had to be used. Strong haptic feedback has been used to reinforce the contrast between the two editing modes, selection and movement.

With the judicious use of well chosen gestures combined with spatial sound, we have demonstrated that a direct manipulation interface can be designed in the auditory domain, one that is typically thought not to be suited to this approach.

Acknowledgements

The work on gestures was funded as part of a DPhil by the UK Engineering and Physical Sciences Research Council (EPSRC) studentship grant 93313798. The work on the Mathtalk program was funded by

EPSRC studentship grant 91308897. The Maths project is funded by the European Union's Tide initiative project number TP1033.

References

Ahlstrom, B., S. Lenman, and T. Marmolin (1992). Overcoming touchscreen user fatigue by workplace design. In *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems – Posters and Short Talks*, Short Talks: Make it Work!, pp. 101–102.

Baudel, T. and M. Beaudouin-Lafon (1993). CHARADE: Remote control of objects using free-hand gestures. *Communications of the ACM 36*(7), 28–35.

Buxton, W. (1986). There's more to interaction than meets the eye: Some issues in manual input. In D. A. Norman and S. W. Draper (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction*, pp. 319–337. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Crispien, K. and K. Fellbaum (1995). Use of acoustic information in screen reader programs for blind computer users: Results from the TIDE project GUIB. In I. P. Porrero and R. P. de la Bellacasa (Eds.), *The Curopean Context for Assistive Technology*, Amsterdam, pp. 306–311. IOS Press.

Crystal, D. (1975). *The English tone of voice.* Oxford University Press.

Edwards, A. D. N. (1991). *Speech Synthesis: Technolgy for Disabled People.* Paul Chapman.

Edwards, A. D. N. and R. D. Stevens (1994). Une interface multimodale pour l'accèss aux formules mathématiques par des élèves ou étudiants aveugles. In *Comme les Autres: Interfaces multimodales pour handicapés visuels, Special number 1*, Université Pierre et Marie Curie, B23, 9 Quai Saint-Bernard, 75252, Paris Cedex 05 and ANPEA (ISSN 0010-2520), pp. 97–104. INSERM.

Fels, S. and G. Hinton (1995). Glove-TalkII: An adaptive gesture-to-formant interface. In *CHI'95 Conference Proceedings.*

Fels, S. S. and G. E. Hinton (1993). Glove-Talk: A neural network interface between a data-glove and a speech synthesizer. *IEEE Transactions on Neural Networks 4*(1), 2–8.

Foster, S., E. Wenzel, and R. Taylor (1991, Oct). Real-time synthesis of complex acoustic environments. In *IEEE workshop on applications of signal processing to audio and acoustic.* New Paltz, N.Y.

Frohlich, D. M. (1993). The history and future of direct manipulation. *Behaviour and Information Technology 12*(6), 315–329.

Garnham, A. (1989). *Psycholinguistics: Central Topics.* Routledge, London.

Haber, R. N. and L. R. Haber (1981). Visual components of the reading process. *Visual Language 15*, 147–182.

Harling, P. and A. Edwards (1995). Hand tension as a gesture segmentation cue. Unpublished paper.

Harling, P. A. (1993, March). Gesture input using neural networks. BEng project report, Department of Computer

Science, University of York, Heslington, York, YO1 5DD, UK.

Hutchins, E., J. Hollan, and D. Norman (1986). Direct manipulation interfaces. In D.A.Norman and S.W.Draper (Eds.), *User Centered System Design*, pp. 87–124. Hillsdale, New Jersey: Lawrence Erlbaum.

Kirshner, D. (1989). The visual syntax of algebra. *Journal for Research into Mathematics Education 20*(3), 274–287.

Larkin, J. H. (1989). Display-based problem solving. In D. Klahr and K. Kotovsky (Eds.), *Complex Information Processing*. Lawrence Erlbaum: Hillsdale New Jersey.

Moore, B. (1989). *An Introduction to the Psychology of Hearing (2nd Edition)*. Academic Press, London.

O'Malley, M. H., D. R. Kloker, and B. Dara-Abrams (1973). Recovering parentheses from spoken algebraic expressions. *IEEE Transactions on Audio and Electroacoustics AU-21*, 217–220.

Pitt, I. and A. Edwards (1991, August). Navigating the interface by sound for blind users. In D. Diaper and N. Hammond (Eds.), *People and Computers VI Proceedings of HCI'91*, pp. 373–383. Cambridge University Press.

Sears, A. (1991). Improving touchscreen keyboards: Design issues and a comparison with other devices. *Interacting with Computers 3*(3), 253–269.

Sellen, A. J., G. P. Kurtenbach, and W. A. S. Buxton (1992). The prevention of mode errors through sensory feedback. *Human-Computer Interaction 7*(2), 141–164.

Shaw, E. A. G. (1974). The external ear. In W. D. Keidel and W. D. Neff (Eds.), *Handbook of Sensory Physiology*, pp. 455–490. Springer Berlag.

Shneiderman, B. (1982). The future of interactive systems and the emergence of direct manipulation. *Behaviour and Information Technology 1*, 237–256.

Shneiderman, B. (1983). Direct manipulation: A step beyond programming languages. *IEEE Computer 16*, 57–69.

Stevens, R. D. and A. D. N. Edwards (1993, January). A sound interface to algebra. In *Proceedings of the IEE Colloquium on Special Needs and the Interface*. IEE Digest no. 1993/005.

Stevens, R. D. and A. D. N. Edwards (1994). Mathtalk: The design of an interface for reading algebra using speech. In W. L. Zagler, G. Busby, and R. L. Wagner (Eds.), *Computers for Handicapped Persons: Proceedings of ICCHP '94, Lecture Notes in Computer Science 860*, pp. 313–320. Berlin: Springer-Verlag.

Stevens, R. D., P. C. Wright, and A. D. N. Edwards (1994). Prosody improves a speech based interface. In D. England (Ed.), *Ancillary Proceedings of HCI'94*. London: British Computer Society.

Streeter, L. A. (1978). Acoustic determinants of phrase boundary representation. *Journal of the Acoustical Society of America 64*, 1582–15:1 92.

Strybel, T. Z., C. L. Manligas, and D. R. Perrott (1992). Minimum audible movment angle as a function of the azimuth and elevation of the source. *Human Factors 34*(3), 267–275.

Sturman, D. J. (1992). *Whole-hand Input*. PhD thesis, M.I.T.

Trevarthen, C. (1986). Form, significance and psychological potential of hand gestures of infants. In J.-L. Nespoulous, P. Perron, and A. R. Lecours (Eds.), *The Biological Foundations of Gestures: Motor and Semiotic Aspects*, pp. 149–202. Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Uras, C. and A. Verri (1995, 26-28 June). Hand gesture recognition from edge maps. In M. Bichsel (Ed.), *Proceedings of the International Workshop on Automatic Face- and Gesture-Recognition*, pp. 116–121.

Wenzel, E., S. Foster, F.L.Wightman, and D.J.Kistler (1989, October). Realtime digital synthesis of localized auditory cues over headphones. In *IEEE workshop on applications of signal processing to audio and acoustics.* New Paltz, N.Y.

Wenzel, E., F. Wightman, and D. Kistler (1991). Localization with non-individualized virtual display cues. In *CHI'91 conference Proceedings*, pp. 351–359. ACM Press, Adison Wesley.

Hand shape	Movement	Command
Pointing	none	Mark
	arc (left to right)	Go to end of expression
	arc (right to left)	Go to beginning of expression
	horizontal movement in expression space	Read
	tapping	Open Hidden Object
Grabbing	none	Select object
	open hand	Release object
	arc (left to right)	Go to end of expression
	arc (right to left)	Go to beginning of expression
	horizontal movement in expression space	Read
	throw over shoulder	Delete
Open hand	left to right sweep	Read whole expression
	vertically down	Mute
Squeeze	from open hand to closed hand	Close hidden expression

Table 2: Summary of the gestures used in Mathgrasp