1 Introduction

The primary goal of this work is to explore the possibility of using visual interpretation of hand gestures as a device to control a general purpose graphical user interface (GUI). There are two sides to the problem: the design of a vision system that can interpret hand gestures in a realistic setting, and the design of an interface that makes good use of them. At some point these problems must be addressed together. Useful gestures exist which can not be reliably recognized, and without a grounding in a realistic interface the gesture recognition system could easily end up tilting at windmills. By examining both aspects of the problem at once, this work attempts to gain a better understanding of the role hand gesture can play in a user interface.

The centerpiece of the thesis is a system that allows a user to manipulate windows and menus in a commercial GUI using hand gestures. The vision system provides reliable recognition of gestures on a modest PC platform in a typical office environment. The interface goes beyond simply replacing the mouse with finger pointing, and makes use of the ability of the hand to form meaningful poses and perform complex motions. It relies on the constraints provided by the environment, the task and the nature of human gesticulation.

The design of the recognition system will be described and discussed in detail. The performance of each component of the system will be examined. Areas where performance can be improved will be pointed out.

The design and performance of the interface will also be examined. The results show the potential of gesture as an interface device. Selection times for large objects are at least as good as with a mouse. Basic window manipulation operations are easy to perform with gestures that suit the task. More complex operations can be performed by
menu selection. The results also show where potential problems with the approach lie, and where more work is needed. Modeling the selection time results suggests that noise in the cursor position must be reduced to improve selection performance. Menus must be redesigned to suit the inherent characteristics of free-hand pointing. Either memory aids must be developed to help people recall the gestural commands or the interaction must be carefully designed to be easy to remember. The thesis ends by reviewing the inherent characteristics of gesture that are important when using it for these types of tasks, discussing alternative interaction techniques that are well suited to gesture, and describing future directions for interface development.

1.1 Why Gesture

Humans naturally use gesture to communicate. It has been demonstrated that young children can readily learn to communicate with gesture before they learn to talk [AG96]. Adults gesticulate in many situations, to accompany or substitute for speech, to communicate with our pets, and occasionally to express our feelings toward uncooperative machines.

The roots of gestural communication become obvious from a few observations. Watching a human baby grow, it is easy to see that we learn much more about the world by manipulating it than by simply observing. Once a child has learned that it can manipulate the world, it is not content to sit passively in its parents arms for long. Only after an object is turned over and lifted, dropped, disassembled and tasted does its appearance begin to substitute for the hands-on experience. It follows that the way we think about objects is closely related to our manipulation of them.

Communication, being an outward expression of our thinking, must also have close ties to manipulation. Evidence for this can be seen when adults speak to each other about an object — you will often see them gesture as if they were manipulating it. Quek [Qu93] and others have observed that this type of free-form “natural” gesticulation, if used repetitively, quickly becomes iconic — that is, specific hand motions take on symbolic meanings.

All this points to the conclusion that gesture is a natural and deeply rooted part of our communication tool kit, yet current interface technology does not take advantage of it, instead using limited devices that only roughly approximate the things our hands evolved to do. This situation was not a conscious design decision, but results from the evolution
of interface technology. For example, until recently, limited CPU power has dictated the complexity of interface devices.

Consider pointing. Pointing at an object is one of the most natural of all gestures, and current GUIs rely heavily on it, yet current interface devices offer only a coarse approximation to pointing. A mouse is one of the best pointing devices yet developed, superior to devices like a joystick or a tablet for many tasks [CEB87], but using it is very different from natural pointing. Being a relative positioning device, the user must first locate the cursor on the screen to determine how the mouse should be moved. This often requires moving the cursor in circles to help find it among the clutter on the screen. Pointing is not direct, a translation is needed between a position on the desk and a position on the screen, which can be difficult for some users to master\(^1\). There are also numerous practical problems, including the dirt that always seems to interfere with ball-driven mice, and the desk space mice require.

The interfaces that have co-evolved with mice have limitations that result from the limitations of the device itself. The only way mice can indicate action in most systems is by some variation of a click. Since the different types of clicks (click, double-click, click-and-drag, etc.) do not contain a great deal of information about the nature of the action the user wants to perform, the location where that click is performed is very important to its interpretation. Thus most interfaces use numerous small active regions on the screen, requiring the user to perform repeated fine positioning operations and using up space that could be better used. Other problems include the omnipresent mouse pointer which always seems to cover some critical information on the screen.

A concrete example will illustrate how a current GUI can become useless, even for an experienced user. While commuting four hours each day on the train, I would often use my laptop to make use of the time. It had a traditional mouse and window interface controlled by trackball. The problem was that the small screen and the motion of the train made it very difficult to find the cursor and control its position with any degree of accuracy. The difficulty of finding the cursor, finding the command region and aligning the two would often make even the simplest tasks a chore. I ended up abandoning window-based interfaces in favor of keyboard commands.

The problem was not with seeing the object I wanted to manipulate — the windows and icons were large enough to locate easily — rather with finding the cursor and

\(^1\) This is based on the author's observations while teaching many people try his Apple Macintosh in the very early days of mouse and window systems.
positioning it within the small control regions on the object. This experience pointed out how much a GUI relies on accurate positioning, and how quickly usability deteriorates if anything interferes with the ability to do that. By relying heavily on one ability, the interface becomes brittle.

The flexibility of hand gestures will make it possible to reduce that reliance. Instead of selecting small control boxes, a user can select the much larger object itself, and perform some gesture indicating how it should be manipulated. In a sense, gesture offers an interface modality midway between keyboards and speech on one side, which control the machine via symbolic commands, and pointing devices such as mice, joysticks and touch screens on the other. The interaction can be more symbolic than the pure dietetic devices, but allows spatial and sub-lingual interactions that are very difficult using verbal or typed commands. This may allow it to take advantages of both styles of interaction, and so expand the capabilities of the interface.

1.1.1 How should gesture be used?

The previous paragraphs have argued that hand gesture recognition may be useful in some hypothetical situations, but is it really worth pursuing gesture in the context of current interface technology? Most users work in a stable, well-lit office, not a moving train. They have a large clear screen and ample desk space, where mature technologies like the mouse serve very nicely. Indeed, others have suggested that control of a windowing interface is not a good application of gesture recognition technology [St92].

Most gesture recognition research has focused on its role in some type of immersive environment, radically different from the interfaces in use today. In contrast, this work is based on the assumption that the next generation user interface is unlikely to be a leap into virtual worlds, rather an incremental step from current technology. It will incorporate the best aspects of today's interfaces augmented by new tools and techniques.

There is much to be gained from extending today's GUIs, rather than scrapping them in pursuit of a new ideal. The tasks for which we use our machines will evolve and grow, but they are not likely to suddenly be replaced, so the metaphors and interaction methods that have proven useful for those tasks are unlikely to suddenly be replaced either. Researchers have gained considerable understanding of current GUI interaction techniques. Programmers have become good at making them comfortable for the user, while still powerful and flexible. Users have developed considerable expertise with both the skills and the concepts they use. It makes sense to retain as much of that knowledge
as we can, so long as we do not limit the growth of the field. It will also lead to greater user acceptance if the interface is a logical extension of existing methods.

Incorporating hand gestures into an existing GUI permits exploration of how it meshes with current interface technology and how it compares to current interface devices. Using this knowledge we can determine if future systems can benefit from using gesture, and if so, what changes must be made to accommodate it.

While the interface envisioned here will look in many ways similar to standard GUls of today, this thesis does not attempt to argue that simply using gesture as a direct mouse replacement in a current GUI will give any real advantages. Indeed, many aspects of current interfaces are tuned to complement specifically the capabilities of the mouse and keyboard, and as such are not well suited for gesture. This domain was chosen as a test-bed for several practical reasons outlined in Section 1.3. It will be argued, however, that in the context of an appropriately designed interface, gesture can offer real advantages as an interface modality.

Beyond this thesis, the goal of this work is a “deviceless” user interface, similar in appearance and function to today's GUls, but using no mechanical devices to interact with the user. Instead, interaction uses integrated speech and gesture commands to eliminate physical interface devices that limit the potential of interface technology. Ultimately, this will make possible a common interface across devices as diverse as communal screens that take up large areas of a wall, personal workstations that use the whole desk surface as screen area, or “phone booths” where people away from their desk could access to resources on a global network such as the Internet.

1.2 Why Vision

Vision is essential for any practical gesture recognition system. While mechanical sensors, such as gloves, have the advantage of simplicity, being able to sense the shape of the hand directly, they have a host of other problems including reliability, accuracy, sanitation and encumbrance. Reliability is a concern with any high-tech mechanical device. Sanitation issues with gloves are obvious. Encumbrance becomes a major problem when one considers gesture as an adjunct to traditional means of interaction, rather than a replacement for them. For example a user may wish to both gesture and type in rapid succession, or flip through the pages of a book as they manipulate an on-screen document.
Accuracy is a problem for current mechanical sensing technology. Errors arise from sensitivity to electro-magnetic noise (prevalent around computers and monitors), slippage of the glove on the hand and differences in fit between individuals, as well as the inherent limitations of the technology [Sa89]. There are ergonometric sources of “noise” as well [SZ89]. Sturman and Zeltzer point out in [SZ89] that users consider gestures to be the same if they look similar, without regard to exact joint angles. This emphasizes that gestures are for humans a visual rather than a tactile form of communication. People naturally form hand shapes that are easy to differentiate visually, implying that machines should use vision to differentiate between them as well.

As far as practical considerations, cameras, frame grabbers and DSP chips are becoming more common on workstations for applications like video conferencing and multi-media. This will reduce the need for special hardware for gesture recognition. Machines are also becoming faster at a tremendous rate, and multi-processor personal systems are already a reality. Finally, notice the trend that computers are spending more of their time and resources on the interface than doing “pure” computing. Thus having a machine spend large amounts of time interpreting the user's actions is not as unusual as it would have been a few years ago.

1.3 Scope of problem

Natural gesticulation is often done without conscious thought, allowing expression of a portion of our thought process that is not well communicated verbally [Mc92]. However, much gesture is also done consciously, with the intent to convey information. Eventually machines should be able to interpret both forms. From unconscious gesture, a form of “body language”, they will glean information about our internal state — our sense of urgency, level of irritation, etc. Much like a sensitive human, a machine may eventually be able to sense the user is becoming irritated that the current interaction is not going well, and so change its interaction style to one that may be more effective. Applications such as this are beyond the scope of this work.

Making use of conscious gesticulation is a much more near-term goal. Here the person has information to convey, and so can assist the machine by being clear and precise, even modifying the gestures used. The information to be obtained is generally more concrete so it is easier to evaluate and use. Thus the primary goal is to explore how conscious gesticulation can be used to interact with a machine.
This work extends an existing window system interface to use gesture. Visual gesture recognition is used for basic operations such as selecting, moving and resizing windows. Mouse input is retained for operations not implemented via gesture. Since the goal of this work is not to develop a practical user interface based on gesture, but is rather to explore how a gesture interface should be designed, there has been no attempt to create a complete or ideal set of interactions using hand gestures. The interactions that have been implemented were intended simply to allow gestural control of a range of typical tasks so that various recognition strategies and interaction styles could be explored.

Using an existing windowing system as a test-bed provides several practical advantages. We do not need to design and write an application for the user to manipulate, simply interpret gestures and send commands to the window system. This way the research can focus on the problem of visual gesture recognition rather than the quirks of a new application. By using a real tool, not a toy application we are forced to address problems common in the real world but often ignored in toy domains.

In order to evaluate an interactive system, its response time must be fast enough to allow it to be used in a realistic manner. To achieve real-time response, some thought has been put into using fast and computationally efficient vision algorithms. On the other hand, since the target is an exploration of the field rather than a product, response time issues have not been addressed beyond what is needed to provide a good demonstration. Similarly — while it is a primary design goal that the system be flexible enough to adapt easily to different users, environments, tasks and interaction modes — ease of calibration or user training have not received as much attention as they need.

The philosophy taken in this work has generally been to adapt state of the art vision techniques to solve an existing problem rather than trying to develop new and improved techniques. This implies living within the limitations of these techniques rather than spending long hours trying to perfect them. If segmentation isn't noise free, we work around noise rather than trying to eliminate it. Rather then design an interaction language that can handle error recovery with elaborate back tracking techniques, we give good feedback to the user of what the system is doing and has understood, give the user the ability to back out of a situation, and let the user do error recovery. This approach has allowed the thesis to stay focused on the problem of visual recognition of hand gestures rather than get sidetracked on interesting but difficult problems in computer vision.
1.5 Overview of the Thesis

The next chapter introduces the reader to the current state of the art in the aspects of computer vision and hand gestures recognition that are pertinent to this work.

Chapter 3 describes the system implementation. Each component is described individually and the reasoning behind the design is discussed.

Chapter 4 evaluates the performance of the system. It gives the results of testing on each component as well as task testing on the final interface and the comments of users.

Chapter 5 is a discussion of what was learned from this work. It examines the work as both a computer vision system for recognizing hand gestures, and as a prototype gesture interface. Suggestions are made for how systems can be better designed to recognize hand gestures and how hand gestures should be used in a user interface. The chapter ends with a discussion of how practical systems using gesture might look.

Finally, Chapter 6 has a summary and some concluding r