UNIVERSITY OF SOUTHAMPTON

A framework for research and design of gesture-based human computer interactions

by

Maria Karam

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

November 2006

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Maria Karam

Gestures have long been considered an interaction technique that can potentially deliver more natural, creative and intuitive methods for communicating with our computers. For over 40 years, gestures provided an alternative input mode to the keyboard and mouse interactions for most application domains, employing an array of technologies to control multitude of tasks. But how do we make sense of the expanse of this technique so that we may approach gestures from a theoretical perspective, and understand its role in human computer interactions? Existing research tends to focus on the technology, exploring novel methods for enabling gestures, and the tasks they can afford. However few researchers have approached the discipline with the intent of building a cohesive understanding of gestures and the relationships that exist between the different systems and interactions. In this work, we present a theoretical framework to support a systematic approach to researching and designing gesture-based interactions. We propose four categories —physical gestures, input devices, output technologies, and user goals —as the basis from which the framework extends. Each category is defined in terms of manipulatable parameters, and their affect on the user experience. Parameters can be tested using empirical experiments, and amended using qualitative methods. The framework is intended for use as a tool to guide research and design, and presents a structure for providing a theoretical understanding of gesture interactions. Our research began with a review and analysis of the gesture literature, preceded by a series of studies and experiments, which lead to the development of the theoretical framework. This thesis presents a detailed discussion of the qualitative and quantitative research that led to the development of framework, its structure and components, and examples of its application towards a theoretical approach to research an design of gestures for human computer interactions.

Contents

List of Figures

List of Tables

Nomenclature

HCI: Human Computer Interaction
Ubicomp: Ubiquitous Computing
SoD: Science of Design
Multimodal: More than one input mode
used either at the same time, or individually.
GT: Grounded Theory
QPR: Qualitative, positivists research methods
DOF: Degrees of freedom
WoZ: Wizard of Oz methodology
PDA: Personal Digital Assistant
CSCW: Computer Supported Collaborative Work

Acknowledgements

I'd like to thank monica schraefel for introducing me to the concept of social responsibility in research, which motivated my interest in academics, and for giving me her trust, and the opportunity to pursue a career in academics. I would also like to thank Jonathon Hare for his steady support, insights into computer vision, and the wonderful iGesture system, and Paul Lewis for his support in my work. I would like to thanks Chris Gutteridge for his multi-faceted and magical assistance in helping me deal with the unforeseen problems that would arise throughout my degree and Chris Bailey, who helped me settle into my role as a PhD student. Also to Max Wilson for his never-faltering ability to provide logic where I only saw chaos and Nick Humphrey, who introduced me to surge radio and made life, the universe and everything possible. To Mike Luck, who's support and wisdom rescued me from many moments of insanity during my degree. Scott McCrickard, Adrian Pickering and Terry -Dj- Payne for their confidence in my work, and Paul Groth for his hours of consultation and tolerance. Thanks to Jason Lee and Travis Rose for their work on the Virginia Experiments. Also to Serena Raffin, Steve Munroe (Spanking Yoda), Miranda Stecyk, Steven Indrigo, Patrick Martinek, Steve Willett, and Sunny Mills for their enduring friendship throughout these challenging years. A special note to The Beacon of fun, Sebastian Francois, Mischa Tuffield for providing me with enough insanity to keep me sane. Mike O Jewell - music Guru, Martin Szomszor, Dave Challis for the thrill of playing with Spanking Yoda. And to Jane Morgan for all her help, along with the rest of the administration staff. Also to Maira Rodrigues, Claudia Di Napoli, Simon Goodall, Thanyalak Maneewatthana, Lester Gilbert, Gary Wills, Ashley Smith, Luc Moreau, Les Carr, Paul Lewis, Hugh Glaser, and Susan Davies for their help with movies, providing equipment, and for their support. I can't forget Wendy Hall, Nick Jennings, Alan Dix for being such good leaders in their field. I must also acknowledge Rain, who gave everything up when I started this degree and who has never left my side, if only in spirit. Finally, to Robert Indrigo, for making everything I do in life so worthwhile!

To Robert Indrigo, Nora Sakr and Family, Greg Karam and Family, and Rain

Chapter 1

Introduction

"To understand the heart and mind of a person, look not at what he has already achieved, but at what he aspires to do."

Kahlil Gibran

1.1 Gestures and Human-Computer Interactions

Gestures have long been considered a promising approach to enabling a natural and intuitive method for human-computer interactions for myriad computing domains, tasks, and applications. The first gestures that were applied to computer interactions date back to the PhD work of Ivan Sutherland (Sutherland, 1963), who demonstrated Sketchpad, an early form of stroke-based gestures using a light pen to manipulate graphical objects on a tablet display. This form of gesturing has since received widespread acceptance in the human-computer interaction (HCI) community, inspiring the stroke-based gesture interactions commonly used for text input on personal digital assistants (PDAs), mobile computing, and pen-based devices (Buxton et al., 1983; Cohen et al., 1997). Since then, the notion of using gestures to facilitate a more expressive and intuitive style of computer interactions has gained popularity among researchers seeking to implement novel interactions with computers. Gloves augmented with electronic motion and position sensors were developed to enhance interactions with virtual reality applications, enabling users to manipulate digital objects using natural hand motions (Sturman et al., 1989; Wexelblat, 1995; Quek, 1994) and polhemus sensors tracked arm movements for controlling large screen displays from a distance, presented by Bolt (1980) in the "Put That There" system. By the mid 1980s, computer vision technology was gaining popularity within the computing sciences, however it was not until the early 1990s that Freeman & Weissman (1995) first demonstrated a vision-based system that enabled gestures to control the volume and channel functions of a television. While this work represented a new direction of perceptual, device-free gestures, computer-vision interactions to date, remain a technique restricted to laboratory studies.

1.1.1 Vision-Enabled Gestures

Although computer vision is widely discussed in the literature as a method for creating more natural gesture interactions with computers, one possible explanation for this slow uptake as an interaction technique is accuracy; that is, perceptual input devices may not provide sufficient control for real-world interactions. But while we do see a more significant uptake in gestures with direct-input devices such as a mouse (Moyle & Cockburn, 2002), stylus (Cohen et al., 1997; Ou et al., 2003), or electronic gloves (Goza et al., 2004), our research considers what level of accuracy would be required before computer-vision gestures can be a viable interaction technique for real-world scenarios. In addition, with the plethora of research on gestures in the literature, we found much ambiguity associated with the term *gesture* and its intended meaning within interaction research. In addition, there are currently no methods in place for determining when gestures would be an appropriate interaction mode, nor for which applications gestures would be best suited to, or how they can best meet users goals. For example, while computer vision gesture research claims it provides a more natural style of interaction, there is no method to define or describe the characteristics of the interaction, or how they affect the user's perception of *natural* in this context. In this work, we consider a move towards gaining a theoretical approach to understanding and designing gesture-based computer interactions.

1.1.2 Background

Our research into gestures was initially motivated by work on the mSpace project (schraefel et al., 2003), where we investigated alternative interaction techniques to support more flexible computer interactions with visual displays. We considered vision-based gestures as a natural choice for its potential to create a more flexible interface for an mSpace browser and for the available resources and collaborations possible within field of computer-vision research in the Intelligence, Agents, Multimedia Group at the University of Southampton.

A comprehensive literature review was conducted on gesture-based interaction research, revealing several important issues that could be addressed. First, with such a large body of literature, it was not clear how to organise the research and access the relevant information for our intended purpose of designing a gesture interface. Second, while the literature contained many systems that implemented vision gestures, most were point designs (Hinckley et al., 1998; Beaudouin-Lafon, 2004): novel interaction techniques that are developed and demonstrated in labs, but that lack the fundamental models and methods required to extend this knowledge beyond the current implementation.

1.1.3 Motivation

While our initial goal was to develop a vision-based gesture system, we wanted first to investigate why perceptual gestures had not yet experienced significant uptake in everyday computing domains, however computer voice interactions were included as standard features in both the Mac OSx and Windows operating systems. Our investigation of the existing research led us to make the following assertions:

1. It is evident that the term *gesture* is used to refer to an expansive range of interactions enabled through a variety of input technologies, devices, and strategies (including computer-vision, data gloves, or touch screens), to control tasks in application domains such as virtual reality, robotics, pervasive, and ubiquitous computing. However, there exists no single theoretical perspective that can support a common discourse when considering gestures as a field of interaction techniques.
2. As we were unable to find evidence suggesting that a common set of terminology exists to describe different gesture systems and interactions, we posited that a set of categories for classifying gestures could form the groundwork for explicitly consolidating the subject into a single interaction technique.
3. By approaching gestures from the perspective proposed in the classification, we could guide our research activities, directions, and applied methodologies to develop a set of common practices for investigating gesture interactions.
4. With a structure in place for investigating gesture interactions, we would then begin to organise our knowledge into a framework to enable a structured method of understanding gestures, and conducing research and design activities.
5. Through subsequent experiments and studies, we could verify our framework and continue to modify its content and structure towards a more complete understanding of gesture interactions.
6. Finally, we would demonstrate how the framework could provide designers with knowledge about appropriate usage scenarios for gestures before they are implemented and a methodology for designing gestures to enhance interactions.

1.1.4 Objectives

In this work, we aim to present a definitive understanding of the study of gestures as a human-computer interaction technique that can promote a methodological approach to

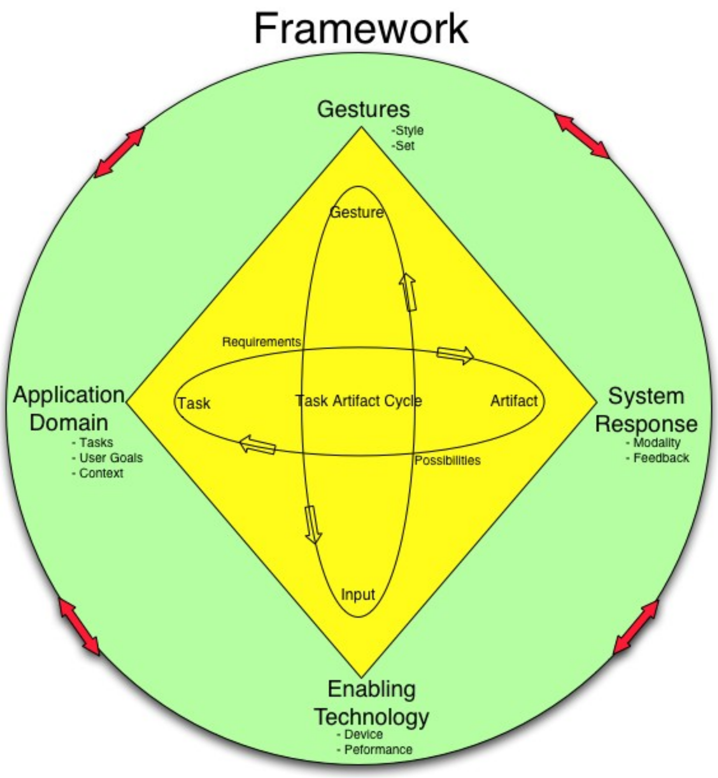

FIGURE 1.1: The diagram represents the organisation and interaction of the theories and elements presented in this work.

designing gesture interactions. First, we identify a set of common, high-level characteristics and categories by which we can begin to compare and contrast gesture interaction research and systems, thus supporting a more cohesive perspective from which to classify the existing research. Second, we present a re-examination of gestures from this new perspective, and execute several experiments and studies that investigate the functional utility of gestures, consider user tolerance for recognition errors within different interaction contexts. Third, we incorporate qualitative and quantitative results from existing gesture research and develop a framework to support research and design activities towards promoting the use of gestures to enhance everyday computing experiences. And fourth, we describe and demonstrate how additional research can contribute to the extension and validation of the framework.

1.2 Thesis Structure

In this section, we present an overview of the topics, and their structure as discussed in this thesis. We describe the structure of this thesis, organised by the research activities that were undertaken over the course of this PhD research.

1.2.1 Chapter 2: A Classification and Analysis of Gesture Interaction Research

In this section we approached gesture interaction from the following perspectives:

1. We conducted a literature survey and an analysis of gesture-based computing research to gain a more cohesive perspective of what the research community refers to as *gesture interactions*.

2. We proposed a set of categories, based on our analysis, to provide a structure for organising the research into the relevant components within each of the categories.

The focus of this research is primarily situated in the domain of computer vision-enabled gesture interactions, where there exists the potential to enhance human-computer interactions by enabling natural, eyes-free interactions with computer systems as demonstrated by many researchers since the early 1980s (Bolt, 1980; Cao & Balakrishnan, 2003; von Hardenberg & Berard, 2001; Crowley & Jolle Coutaz, 2000; Freeman & Weissman, 1995). But still, after more than 25 years of research into vision-based gestures, and the numerous enhancements demonstrated to improve our interactions with computers, gestures are not understood. In this research, we begin to investigate possible reasons as to why this is so. As well, making sense of the volume of gesture-based research that has been conducted in the field of human-computer interactions remains a daunting task given the vast array of applications, technologies, and gestures considered. In an attempt to provide a more cohesive perspective on gestures as a field of computer interaction techniques, we present an approach to systematising research in gesture-based interaction techniques. While there is a body of research that focuses on developing a classification scheme for the different styles of human gesturing, our work addresses gesture interaction systems in their entirety, from the HCI perspective. We conducted a survey and literature review of gesture-based interactions in the computing literature, and codified gesture-based interaction systems into several categories, enabling us to distinguish between systems by addressing elements that included the physical movements required to perform the gesture (gesture style), the nature of the interaction (application domain) how the gestures are recognised (input), and what the intended result will produce (output). We then will discuss our approach to classifying the literature, and how this informed the directions of this research.

1.2.2 Chapter 3: Exploring Functional Utility of Gesture Interactions

With a structure in place for viewing the scope of gesture interaction research, we could begin to address some unresolved issues identified in the literature. We focused on the following points: Semaphoric gestures are one of the most common approaches to gesture interaction reviewed in the literature, and potentially one of the simplest forms to recognise using vision technology. However, Wexelblat (1998) suggest that semaphoric gestures are not a useful method for signalling controls to a computer despite their dominant presence in the literature. Wexelblat also argues that since semaphoric gestures provide roughly only 90% accurate recognition rates, they offer no functional utility over a keyboard, which is capable of 100% accurate recognition. In this chapter, we investigate the functionality of gestures based on several studies we conducted and address the following issues:

1. We identify a potential use for semaphoric gestures in computer interactions.
2. We discuss specific contexts in which gestures provide enhancements to existing interactions.

We conducted interviews and pilot studies with 25 participants from our lab to investigate the functionality of gestures when controlling secondary tasks within the context of multitasking situations (Hare et al., 2005). We tested our hypotheses in an empirical study to determine if semaphoric gestures could offer benefits over a keyboard interaction when controlling secondary tasks in a multitasking situation. Details of this study are presented in Chapter 3, with results suggesting that semaphoric gestures can improve performance in multitasking situations by causing less distraction to the users primary task, leading to faster completion of secondary tasks than with the keyboard input. To extend these results, we next investigated errors in vision-enabled gesture recognition, and their effects on users tolerance.

1.2.3 Chapter 4: Investigating User Tolerance and Performance Issues during Gesture Recognition

Our second empirical study addressed errors in vision-enabled gesture recognition systems in terms of user tolerance levels and the various contexts in which gesture interactions are employed. We designed a study where gestures were used to control secondary tasks in multitasking situations. However, to address the issue of recognition errors, and extend our previous work, we consider the following concepts:

1. The level of recognition error that users will tolerate in gesture interactions.
2. Additional contexts for extending our knowledge about multitasking situations and gestures.

We investigated two different interaction scenarios. The first, described as a desktop scenario reflects the standard computing model, where the interaction occurs exclusively at the desk. The second is described as a ubiquitous computing scenario, where the interaction is distributed using input and output devices located throughout ones environment. We used a participant observation study to assist in the design of our interaction scenario (see Appendix C), to enable us to create a more realistic environment, and to expand the context investigated in our previous chapter. Results revealed a proportionate relationship between the users perceived convenience of using gestures and the level of tolerance exhibited for errors, and an inversely proportionate relationship between error-tolerance and the level of importance placed on an interaction. Details and results of the experiment are discussed in Chapter 4.

1.2.4 Chapter 5: Towards A Framework for Gesture Research and Design Gesture Interaction

We next wanted to determine the best use of the knowledge gained in our previous research to assist with understanding and designing gestures as a computer interaction technique. This chapter presents the following contributions:

1. We develop a framework for organising and comparing empirical knowledge about gesture interactions.
2. We demonstrate how the framework can support a methodological approach to conducting research on gesture interactions.

We developed a theoretical framework to support research and design activities that incorporates existing knowledge, while providing a structure for incorporating future gesture towards building a richer understanding of gestures as an interaction technique. We employed a qualitative approach and used the grounded theory methodology to develop the framework, and an empirical approach to verify its content. An early version of the framework is shown in Figure 1.2. The framework provides a structure that presents a common set of terminology to describe, compare, measure, and design gesture-interaction systems. Each of the four categories or elements of the framework —gesture style, input (enabling technology), application domain and output (system response) —are elaborated on using parameters that describe concepts revealed as having an affect on users during interactions. The intended uses of the framework include guiding research and design activities, incorporating new knowledge, and providing a cohesive structure to support a methodological approach to improving gesture-based interaction systems, designs, and interactions. In this chapter, we present the evolution of the framework, and provide an in-depth discussion of its components.

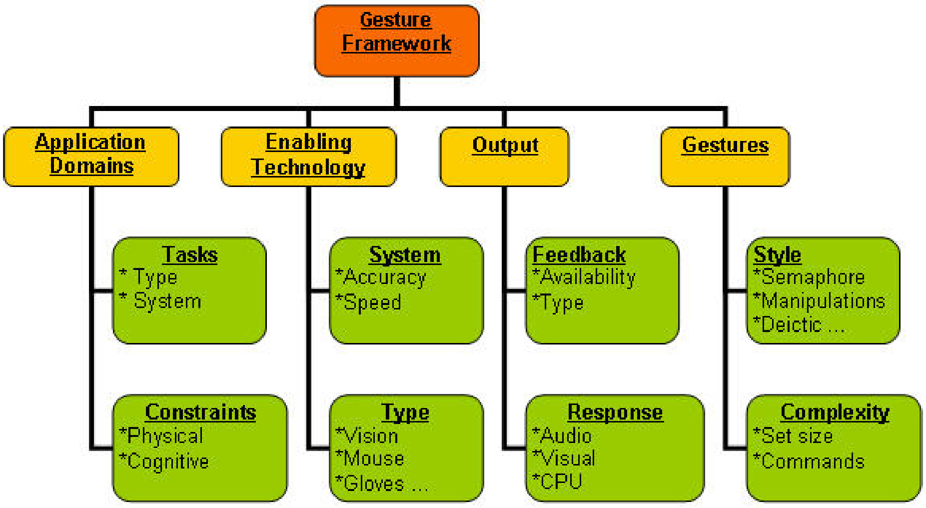

FIGURE 1.2: The diagram presents an early version of the structure of the framework which attempts to reveal specific characteristics of the categories used to classify the research.

1.2.5 Chapter 6: Framework Verification and Extension

This chapter presents additional research and experiments we conducted to verify the framework, increasing its scope using experimental results. We also evaluate the capacity of the framework to predict users interaction behaviour, and its role in suggesting and motivating new research directions. The first study extends results from our experiment on the functional utility of gestures presented in 3, to validate the ability of the framework to predict user behaviour in a different interaction scenario. The second study considers the concept of reflexive feedback, and reveal several additional parameters to increase our knowledge of the feedback parameter within the framework. And finally we explore issues and strategies for group interactions, extending our framework to include the computer-supported collaborative work domain (CSCW).

1.2.6 Chapter 7: Future Work and Conclusions

In this chapter, we discuss several potential directions for this research, including our intended approach to demonstrate the enhancement stage to conducting HCI research and a contribution towards a science of design approach to interaction design. We also address extensions of the framework that involve its restructuring to create an additional element —interaction technique. Here, the category *gestures* would become a sub-category of the main category *interaction technique*, suggesting that the framework may be applied to the design and understanding of other input techniques such as speech recognition, and tangible input for example. We conclude with our summary of the work to date, and our contributions to the field of HCI.

Chapter 2

A Review and Analysis of Gesture Interaction Research

> *"Character cannot be developed in ease and quiet. Only through experience of trial and suffering can the soul be strengthened, ambition inspired, and success achieved."*

Helen Keller

2.1 It Began Here

This chapter presents a literature review, analysis and classification of gesture-based human-computer interaction research. We began this literature review in search of an interaction technique that could better complement the ambient interactions afforded by an mSpace music browser. Studies suggested that users could successfully navigate a music library without a requiring a visual display (schraefel et al., 2003). This motivated our choice to consider vision-enabled gesture interactions, paving the way for the direction of this research. Since we were already working with the iGesture system —a gesture interaction tool developed by Jonathon Hare in the IAM Group for enabling research in computer vision —we quickly became familiar with some of the issues inherent in computer vision technology: mainly due to the difficulties in tracking objects in variable lighting conditions. But after several weeks of working with iGesture, we began to question if there would be any value to an interaction technique with a recognition rate would change with the lighting. However, before we went ahead and made changes to iGesture, we realised that we needed to better understand the users perspective to enable us to make informed improvements, rather than just altering the system. This proved a difficult task, since we found no research to suggest vision gestures were ever used outside of a research lab, nor that provided research on the affect gestures have on

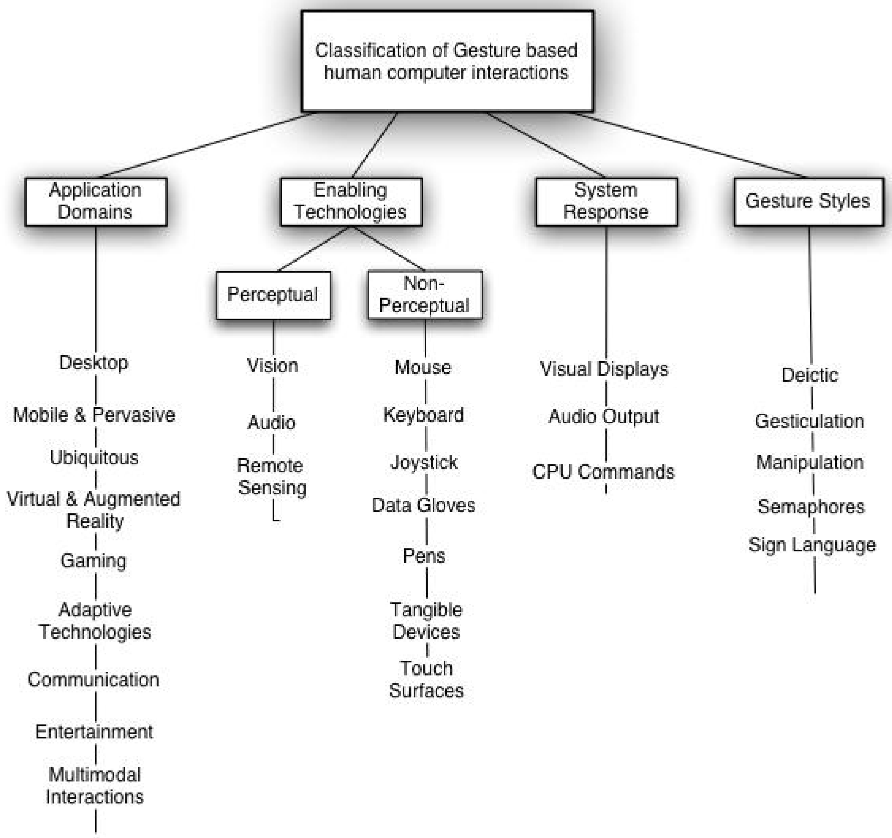

FIGURE 2.1: The diagram shows the organisation of the literature we reviewed in the four categories for classifying the research.

users during interactions. In response, we set out to discover if we could gain a more theoretical perspective of gestures as an interaction technique. In the following sections, we present our literature review, and the resulting analysis which led to our proposed classification of the research. Figure 2.1 presents an overview of the categories we used to organise our classification. Parts of this chapter will be submitted to the Journal ACM Surveys, and a preliminary version of this chapter is presented as a University of Southampton technical report (Karam & schraefel, 2005b).

2.1.1 Approach

We reviewed the large body of research within the domain of gesture interactions to attempt to gain a theoretical understand of the literature and the scope of what exactly researchers consider as gestures. We employed content analysis techniques as our approach to organising the literature towards presenting a cohesive perspective of the research.

Content analysis. Content analysis is a technique for gathering and analysing the content of text typically in the social sciences (Schloss & Smith, 1999). We applied content analysis on our literature review to assist in uncovering some of the common themes and concepts that are addressed in the research. The process began with the open coding technique —where we identified terms, concepts, and issues that were discussed in the literature to gain a more complete understanding of the domain. These concepts were gathered and coded into a set of terms and stored in a table to represent our knowledge of gestures. As new concepts were revealed thorough the review process, our codes were altered to reflect the research, and resulted in the categories presented in this chapter. A complete listing of the literature reviewed, and the content analysis tables are included in Appendix A. Our review led to the following concepts, derived from the literature and coded to include author name, publication title, conference or journal name, and year:

- Gesture style: The physical movements referred to as a gesture.
- Body part: The part of the body or object used to execute a gesture.
- Input technology: Input devices that enable gestures.
- Output: The device output behaviour resulting from an executed gesture.
- Area of research: Stated as the domain or research focus of the publication.
- Evaluation: The type of study —ranging from short user trials to extensive empirical evaluations —described in the publication.
- Problems addressed: Stated in the publication.
- Motivation: Stated in the publication.
- Research focus: Stated in the publication.

Repeated analysis of our coding technique led to a selective coding stage, where we were able to refine the concepts reviewed in the literature and identify four categories to reflect the components common to all gesture interaction systems: Gesture style, application domain, input (enabling technologies) and output (system responses). Figure 2.2 presents an early version of our classification scheme where we began to reveal different systems and their position within the classification. Our literature review and analysis is presented next, organised according to the categories used for the classification.

2.2 Understanding Human Gestures

We begin with a brief history and overview of gestures within the multi-disciplinary field of human-gesturing. Concepts from linguistics, anthropology, cognitive science and

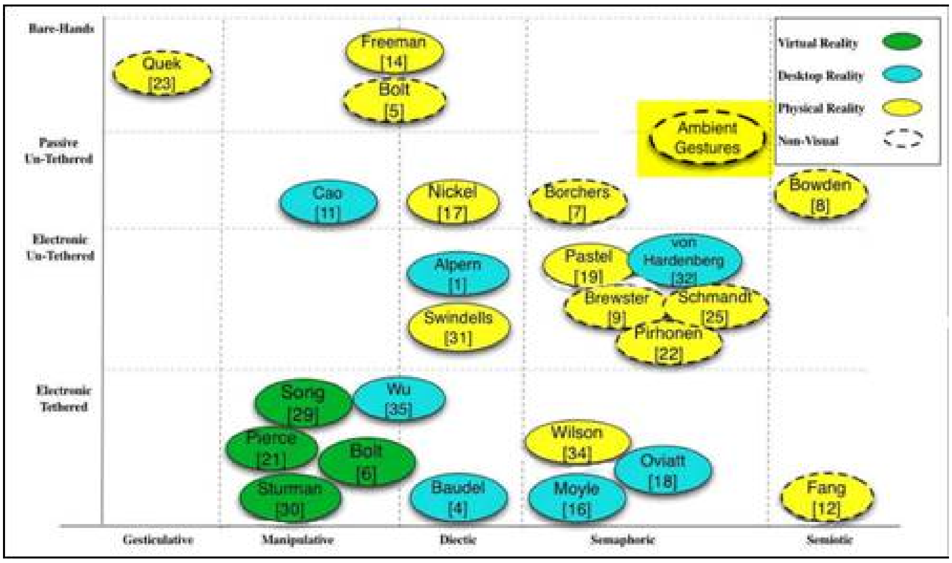

Figure 2.2: The diagram shows some of the first systems we reviewed, and our first attempt at organising the literature into a cohesive structure along these categories.

psychology contribute to this field, and there are many ongoing debates how to classify human gestures. This topic is beyond the scope of our research, but for a detailed discussion about human gestures, we refer to The University of Chicago's Centre for Gesture and Speech Research and the McNeill Lab (ges, 2006). Several researchers in the field of human gesturing have attempted to classify gestures including Ekman, Mespoulos, and Lecours (Wexelblat, 1998), however researchers in the computing sciences often refer to work by Kendon and McNeil for their classifications (e.g. Quek et al., 2002; Eisenstein & Davis, 2004; Kettebekov, 2004; Wexelblat, 1998).

2.2.1 Classifying Gestures

One of the main problems within gesture research is the lack of any commonly used terms for describing the interactions. For example, gesticulations are often referred to as co-verbal gestures, pantomimes or natural gestures (Kettebekov, 2004; Quek et al., 2002; Wexelblat, 1995), while the term natural gestures also refers to bare or free handed gestures (e.g. von Hardenberg & Berard, 2001; Baudel & Beaudouin-Lafon, 1993; Eisenstein & Davis, 2004). In addition, symbolic gestures are also described as iconic or stroke gestures (Kopp et al., 2004; Koons & Sparrell, 1994). The computing literature provides several attempts at classifying gestures including Wexelblat (1998), Kettebekov (2004), Brereton et al. (2003), and Pavlovic et al. (1997). For the purpose of our research, a high-level classification of the gestures was sufficient to enable us to organise the literature, and we refer work by Quek et al. (2002), who proposed a framework that considers manipulations, semaphores and gesture-speech approaches (gesticulation) as the primary focus of existing gesture research. We extend this framework and include deictic (pointing) and language-based gestures (sign language) to reflect their prevalence in the computing literature. We discuss the gesture styles, and provide examples from the literature next.

2.2.2 Deictic Gestures

Deictic gestures involve pointing to establish the identity or spatial location of an object. Although deictic gestures can be similar to the direct manipulation input of a mouse (e.g. Wellner, 1991; Rubine, 1992; Ward et al., 2000; Ou et al., 2003), they also represent a large proportion of the interactions described in the literature. The seminal example is the "Put that there" work by Bolt (1980). Bolt's deictic gestures are used in conjunction with speech input, allowing the user to point at an object on a large screen display while speaking to indicate an action to execute. Additional forms of deictic gestures include pointing within virtual reality displays (Zimmerman et al., 1995), pointing to communicate to collaborators in remote locations (Kuzuoka et al., 1994), pointing to target appliances in smart room environments (e.g. Swindells et al., 2002; Nickel & Stiefelhagen, 2003), identifying objects or windows on desktop applications on the DigitalDesk by Wellner (1991), or in work by Kobsa et al. (1986) to augment speech in communication applications.

2.2.3 Manipulative Gestures

Quek et al. (2002) defines a manipulative gesture for controlling an object as

> applying a tight relationship between the actual movements of the gesturing hand/arm with the entity being manipulated.

While a direct manipulation could include a drag and drop operation using a mouse, a manipulative gesture would involve more complicated interactions requiring interpretation by the computer system, such as manipulations using pen strokes to indicate movements of an on-screen object Rubine (1992), or a physical object such as a camera (Zeleznik & Forsberg, 1999). Manipulations can include two-dimensional (2D) movements input along an x-y axis, as with a mouse, or multi-dimensional movements in space, tracking complex hand motions (Sturman et al., 1989; Zimmerman et al., 1987). We next discuss manipulative gestures citing examples from the literature.

2D manipulations. Gesture manipulations often occur within 2D displays for controlling graphic objects, or windows. The traditional on-screen interaction involves direct manipulations of these objects using a mouse, however a manipulative gesture according to Rubine (1992) differs because of the provision of parameters to the system, indicating the nature of a transformation or relocation of the digitally rendered object. In recent work, Wu & Balakrishnan (2003) demonstrated 2D manipulative gestures for table top surfaces fitted with electronic material. Similar work by Rekimoto (2002) used manipulative gestures drawn from actual table top gestures such as sweeping and isolating groups of objects with the hands to manipulate them. Similar systems can also employ 3D gestures through the use of pressure or weight sensors, to enable more complicated manipulations. We discuss 3D gestures next.

3D manipulations. 3D manipulations of 2D digital-objects often involve an additional element of pressure, weight or velocity through additional sensors to control a graphics display as in a finger painting application that senses the pressure exerted on the display to indicate line thickness (Minsky, 1984). 3D manipulative gestures are also used to identify and transfer digital objects between different devices, as in the Pick and Drop gestures by Rekimoto (1997). We next explore gestures for manipulating tangible objects.

Tangible gestures and digital objects. Tangible objects that are used as computer input are often referred to as gestures. Hinckley et al. (1998) present an interaction where the physical manipulation of a doll's head is mapped onto its on-screen representation of a human brain, and Sinclair & Martinez (2002) consider shaking a tangible cube as a gesture to add labels to an augmented reality display.

Tangible gestures and physical objects. Manipulative gestures can also control physical objects such as robot arms (Goza et al., 2004; Fisher et al., 1987) or vehicles such as wheel chairs (Segen & Kumar, 1998b). Devices can be manipulated to create gestures that indicate file transfers between devices (Hinckley, 2003) or the shaking mobile phones in specific patterns to identify individual devices for use with public display interactions (Patel et al., 2004). Manipulative gestures also include interactions that track physical movement and interpret that input as Midi output, simulating the creation of music using a conductor's baton Borchers (1997). This free-form style of manipulation is also used to create an air-guitar simulation using the iGesture system Hare et al. (2005), or body movements such as dance are interpreted as gestures to control computer output Tarabella & Bertini (2000). These can be considered manipulative gestures since the movements must be interpreted as input (gesture) to create the output (resulting media controls).

2.2.4 Semaphoric Gestures

We refer to the definition of semahporic gestures provided by Quek et al. (2002).

> Semaphores are systems of signalling using flags, lights or arms [Brittanica.com]. By extension, we define semaphoric gestures to be any gesturing system that employs a stylised dictionary of static or dynamic hand or arm gestures...Semaphoric approaches may be referred to as "communicative" in that gestures serve as a universe of symbols to be communicated to the machine.

Semaphoric gestures are frequently discussed in the literature as being one of the most widely applied, yet least used forms of human gesturing Wexelblat (1998); Quek et al. (2002). However, semaphoric gestures are still seen as a practical method of providing distance interactions for smart rooms and intelligent environments (Bolt, 1980; Baudel & Beaudouin-Lafon, 1993; Cao & Balakrishnan, 2003; Lenman et al., 2002b; Wilson & Shafer, 2003; Streitz et al., 1999) and enabling eyes-free interactions (see Chapter 3). Semaphores in their different forms reviewed in the literature are discussed next.

Static vs. dynamic gestures. Semaphoric gestures can include static poses or dynamic movements. For example, joining the thumb and forefinger to form the "ok" symbol is a static pose, while waving a hand is dynamic. Semaphores are symbols that can be executed using hands (Alpern & Minardo, 2003; Baudel & Beaudouin-Lafon, 1993; Rekimoto, 2002; Lee et al., 1998), fingers (Grossman et al., 2004; Rekimoto et al., 2003), arms (Nickel & Stiefelhagen, 2003; Bolt, 1980), heads (Schmandt et al., 2002; Davis & Vaks, 2001), feet (Paradiso et al., 2000b) or objects such as such as a wand or a mouse (Wilson & Shafer, 2003; Baudel & Beaudouin-Lafon, 1993; Moyle & Cockburn, 2002).

Stroke gestures. Stroke gestures, such as those executed using a pen or stylus are also considered semaphores, and include flicking the mouse back and forth to interact with web browsers (Moyle & Cockburn, 2002), or for screen navigation or marking or pie menu selections (Smith & schraefel, 2004; Lenman et al., 2002b; Zhao & Balakrishnan, 2004). Stroke gestures are also used to control avatars (Barrientos & Canny, 2002), desktop computing applications (Segen & Kumar, 1998a; Chatty & Lecoanet, 1996; Wu & Balakrishnan, 2003; Ou et al., 2003; Allan Christian Long et al., 1999; Pastel & Skalsky, 2004) and include Graffiti, and Jot stroke-based languages (Ward et al., 2000; Forsberg et al., 1998; Pirhonen et al., 2002; Rubine, 1992; Cohen et al., 1997).

2.2.5 Gesticulation

Considered one of the most natural forms of gesturing, gesticulation gestures are commonly considered for use in multimodal speech interfaces (Quek et al., 2002; Wexelblat, 1994; Kopp et al., 2004; Bolt & Herranz, 1992; Kettebekov, 2004; Silva & Arriaga, 2003;

Eisenstein & Davis, 2004; Krum et al., 2002). Originally referred to by (Bolt & Herranz, 1992) and Kettebekov (2004) as 'coverbal gestures', a term credited to Nespoulous and Lecours, gesticulation research has recently gained a great deal of attention in the literature and is currently viewed as one of the most challenging areas of gesture research. Gesticulations rely on computational analysis of hand movements in the context of speech. Wexelblat (1995) refers to gesticulations as idiosyncratic, not taught, empty handed gestures that are considered for directive style interfaces. Also referred to as depictive or iconic gestures, gesticulations are intended to add clarity to speech recognition, where a verbal description of a physical shape or form is depicted through the gestures (Koons & Sparrell, 1994; Bolt & Herranz, 1992; Kopp et al., 2004). Iconic gestures and pantomime also fall within this category, however most of the research in this area to date remains theoretical, and far from implementation.

2.2.6 Language Gestures

Sign languages are considered independent from other gesture styles since they are linguistic-based and require the collective interpretation of multiple, individual hand signs that combine to form grammatical structures. While Finger spelling can be considered semaphoric gestures[1], its use in conversational interfaces constitutes an additional level of processing which is more like gesticulation, where systems are required to interpret collections of signs as a meaningful string (e.g. Bowden et al., 2003; Braffort, 1996; Fang et al., 2003; Sagawa et al., 1997), rather than early work by (Zimmerman et al., 1987) and gestures for communicating individual letters for finger-spelling applications.

2.2.7 Multiple Gesture Styles

Many of the systems we reviewed are designed to employ multiple styles of gestures: combining deictic and manipulative gestures (Rekimoto, 1997; Fisher et al., 1987; Sharma et al., 1996), semaphores and manipulations (Joseph J. LaViola et al., 2001; Weimer & Ganapathy, 1989; Sturman et al., 1989; Nishino et al., 1997; Ou et al., 2003; Grossman et al., 2004) deictic and semaphoric gestures (Baudel & Beaudouin-Lafon, 1993; Konrad et al., 2003; Wilson & Shafer, 2003) and semaphoric, deictic and manipulative gestures (Cao & Balakrishnan, 2003; Osawa et al., 2000; Rekimoto, 2002; Wu & Balakrishnan, 2003). FingARTips by Buchmann et al. (2004) demonstrates an augmented reality interaction where gestures are employed to identify and manipulate objects, and to issue commands.

[1]Individual letters used in finger spelling can be considered semaphoric gestures, in that they are signs to be interpreted by the computer. However language gestures require the entire string of letters or words to be interpreted as a linguistic structure so that meaning is provided, rather than being interpreted in isolation as a semaphore.

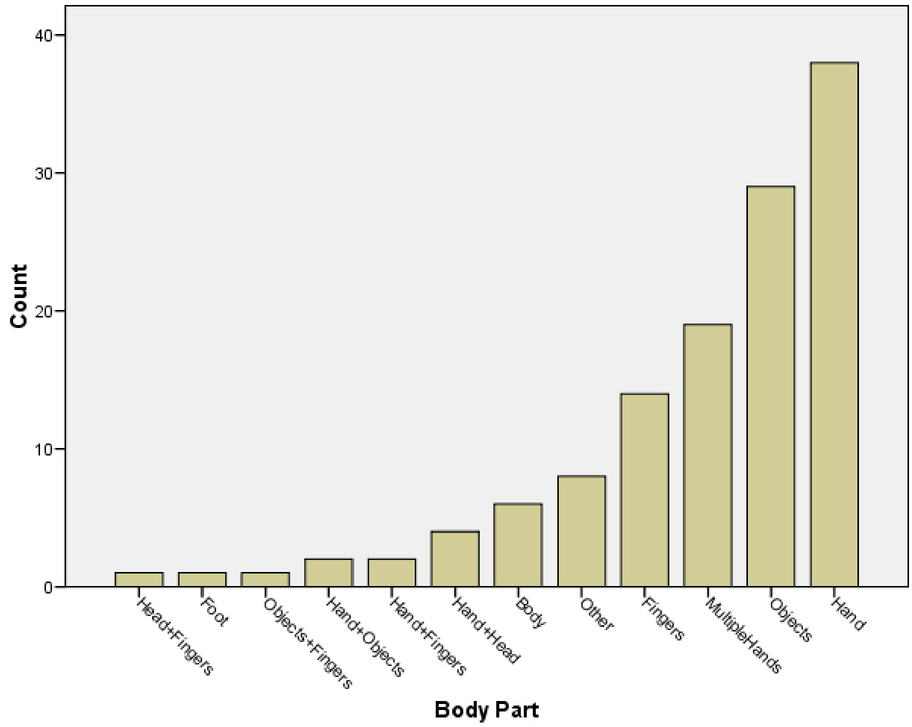

FIGURE 2.3: Gesturing and body parts: The bar graph shows the different body parts or objects, identified the literature, employed for gesturing.

2.2.8 Analysis of Gesture Styles

One of the fastest growing styles of gesture research is in the area gesticulation for communication interfaces (e.g. Kettebekov, 2004; Eisenstein & Davis, 2004; Quek et al., 2002; Kopp et al., 2004), yet we were not able to locate any examples of working systems. This may be possibly due to the challenges that remain within this form of gesture interaction: researchers must first understand the relationship between gesticulation and speech to implement a system, and additional problems of parsing and disseminating individual gestures from continuous set must still be addressed. But while researchers claim that gesticulation can assist in the disambiguation of speech recognition and provide more natural interactions, there is little supporting evidence to suggest that this will occur in the near future.

Although semaphoric gestures are not always considered as natural as gesticulation, they are a reality, and are the least complicated to implement. In addition, they are a practical interaction technique for ubiquitous computing interactions where distance interactions are of interest, and for mobile computing where the eyes-free interactions they afford can support users in multitasking situations Brewster et al. (2003). However, it is not clear if vision gestures can provide enhanced interactions within these domains given their inherent level of inaccuracy in recognition. We investigate this issue in Chapter 3.

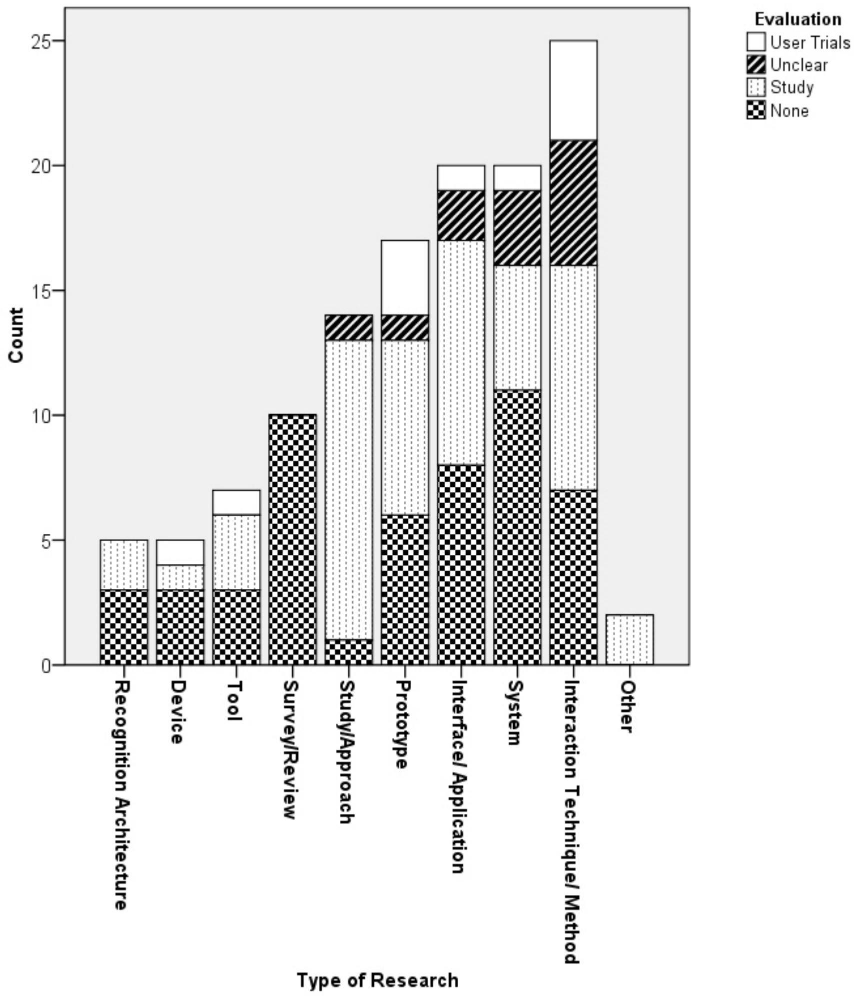

FIGURE 2.4: Evaluations according to the nature of the research reviewed in the literature.

While multiple gesture styles were most commonly referenced in the literature, it may not always be beneficial to use gestures for all tasks within a given application. We next discuss gestures and the application domains that employ them.

2.3 Application Domains

In this section, we present an overview of the application domains that employ gesture interactions. Figure 2.5 presents a summary of the domains addressed in our literature review, discussed next.

2.3.1 Virtual Reality

Gestures for virtual and augmented reality applications have experienced one of the greatest levels of uptake in computing. Virtual reality interactions use gestures to enable realistic manipulations of virtual objects using ones hands, for 3D display interactions Sharma et al. (1996) or 2D displays that simulate 3D interactions (Gandy et al., 2000). We identified three sub-categories of virtual reality applications where gestures are primarily employed: non-immersed interactions, where the users are not represented within the virtual world, semi-immersed interactions, where a user is represented in the virtual world as an avatar, and fully-immersed interactions where the user interacts from the perspective of being inside the virtual world.

Non-immersed interactions. Interactions with gestures can involve navigating in and around a 3D or virtual display from an observers perspective (e.g. Segen & Kumar, 1998a; Nishino et al., 1998). Osawa et al. (2000) used hand gestures to arrange virtual objects and to navigate around a 3D information space such as a graph, using a stereoscopic display. These interactions do not require any representations of the user and user gestures primarily for navigation and manipulation tasks.

Semi-immersed interactions and avatars. An early example of a semi-immersed interaction is seen in Videoplace by Krueger et al. (1985), where users could interact with their own image projected onto a wall to create the virtual world display. Semi-immersed virtual interaction often use avatars to represent users, and work by Barrientos & Canny (2002) uses stroke gestures to control avatar expressions and movements. More complicated avatar interactions include tracking full body gestures to model and control an avatars movements within a virtual world (e.g. Thalmann, 2000; Maes et al., 1997; Lee et al., 1998). Joseph J. LaViola et al. (2001) creates a semi-immersed interaction using sensors embedded in the floor and in the user's shoes to detect their motion and location to map onto the virtual world.

Fully-immersed interactions and object manipulations. Fully-immersed interaction require DataGloves or other sensing devices that can track movements and replicate them and the user within the virtual world (e.g. Fisher et al., 1987; Song et al., 2000; Pierce & Pausch, 2002; Nishino et al., 1998). Zimmerman et al. (1987) and Sturman et al. (1989) presented work towards developing a system for manipulating virtual objects, where the user's hand movements are reproduced in the virtual world. Sturman identified three different types of interactions for whole hand gestures; direct manipulations, where a user reaches into a simulation to manipulate objects, abstracted graphical input such as pushing buttons, and movement, which is interpreted as a stream of tokens. While we can refer to these as manipulative or semaphoric gestures, their use within virtual reality applications has continued to grow throughout the 90's (e.g. Weimer & Ganapathy, 1989; Bolt & Herranz, 1992; Koons & Sparrell, 1994; Wexelblat, 1995; Nishino et al., 1997) and remains as one of primary modes for interacting in immersed virtual interactions Pierce & Pausch (2002).

2.3.2 Augmented Reality

Augmented realty applications often user markers, consisting of patterns printed on physical objects, which can more easily be tracked using computer vision, and that are used for displaying virtual objects in augmented reality displays. Buchmann et al. (2004) uses such markers in combination with bare hand movements to recognise gestures for selection and manipulation tasks in an AR display.

2.3.3 Robotics and Telepresence

Telepresence and telerobotic applications are typically situated within the domain of space exploration and military-based research projects. The gestures used to interact with and control robots are similar to fully-immersed virtual reality interactions, however the worlds are often real, presenting the operator with video feed from cameras located on the robot (Goza et al., 2004). Here, gestures can control a robots hand and arm movements to reach for and manipulate actual objects, as well its movement through the world.

2.3.4 Desktop and Tablet PC Applications

In desktop computing applications, gestures can provide an alternative interaction to the mouse and keyboard (e.g. Iannizzotto et al., 2001; Stotts et al., 2004). Many gestures for desktop computing tasks involve manipulating graphic objects (e.g. Bolt & Herranz, 1992; Buxton et al., 1983), or annotating and editing documents using pen-based gestures (Cohen et al., 1997). Smith & schraefel (2004) also use pen gestures, where circular motion creates a radial-scrolling effect for navigating through documents, while Lenman et al. (2002b) make marking menu selections using stroke gestures. Mouse gestures are also used for various applications including web browsing tasks (Moyle & Cockburn, 2002). But most of the gestures that are seen in desktop applications employ direct input devices such as a pen or mouse (e.g. Rubine, 1992; Kjeldsen & Kender, 1996; Cohen et al., 1997; Dannenberg & Amon, 1989; Henry et al., 1990). Similar gestures are commonly used for tablet computers in specialised applications for air traffic control rooms (Chatty & Lecoanet, 1996), adaptive technology (Ward et al., 2000) and musical score editing (Forsberg et al., 1998). Gestures using non-direct input devices for desktop computing also include nodding to respond to dialogue boxes (Davis & Vaks, 2001), however most of the applications for desktop domains use the standard direct-input devices. We discuss input in Section 2.4.

Graphics and drawing applications. One of the first applications for gestures was in graphics manipulation from as early as 1964 (Sutherland, 1963; Teitelman, 1964; Coleman, 1969). The gestures consisted of strokes, lines, or circles used for drawing, controlling applications, or switching modes (Buxton et al., 1983; Rhyne, 1987). Buxton et al. (1985) presented similar interactions with touch screens and tablet computers using fingers as well as pens. Minsky (1984) also uses pens, however introduces pressure sensitive interactions where users can change the thickness of lines by changing the pressure exerted on the surface.

2.3.5 Computer Supported Collaborative Work (CSCW)

Gestures are used in CSCW applications to enable multiple users to interact with a shared display, using a variety of computing devices such as desktop or tabletop displays (Wu & Balakrishnan, 2003; Rekimoto, 2002) or large screen displays (Cao & Balakrishnan, 2003; von Hardenberg & Berard, 2001). Notes and annotations can be shared within groups using strokes both locally or for remote interactions (Wolf & Rhyne, 1993; Gutwin & Penner, 2002; Stotts et al., 2004). Annotations can be transmitted using live video streams, to enable remote collaborations between students and instructors (Kuzuoka et al., 1994; Ou et al., 2003).

2.3.6 Ubiquitous Computing and Smart Environments

Early work on gestures demonstrated how distance interactions for displays or devices could be enabled within smart room environments (Bolt, 1980; Krueger et al., 1985). However, it was not until several years later that Weiser (1993) described his vision of ubiquitous computing and gestures gained popularity as an interaction mode in this domain. Zimmerman et al. (1987) used devices to sense human presence and position within a room to enable non-contact interactions with computer devices, and it was not until 1994 that gestures using computer vision were considered for use as a television controller by Freeman & Weissman (1995). Additional examples of smart room interactions use gestures to signal the transfer of data between different devices (Rekimoto, 1997; Swindells et al., 2002). As smart room technologies became more sophisticated, so did the notion of using perceptual style input to enable gestures to control smart room applications Crowley & Jolle Coutaz (2000), which included controlling lights, entertainment units or appliances (e.g. Wilson & Shafer, 2003; Fails & Olsen, 2002; Lenman et al., 2002b; Nickel & Stiefelhagen, 2003), and interactions with large screen displays (e.g. Paradiso et al., 2000a; von Hardenberg & Berard, 2001).

2.3.7 Tangible Computing

One of the first systems that considered tangible interactions was work by Fitzmaurice et al. (1995) on Bricks. Bricks are physical objects that can be manipulated to control corresponding digital objects. By 1991, Wellner (1991) used pointing gestures to identify numbers on a printed page as input to a computer. More novel interaction involving gestures include moving, bumping, or squeezing devices to invoke commands and communicate information about the user (e.g. Hinckley, 2003; Harrison et al., 1998). Tangible interactions also involve manipulating dolls embedded with sensors to communicate emotion for 3D virtual games (Paiva et al., 2002) or smart-fabric lined objects that cause reactions for creatures in artificial life applications (Schiphorst et al., 2002).

2.3.8 Pervasive and Mobile Computing

Gestures can enable eyes-free interactions with mobile devices that allow users to focus their visual attention on their primary task (e.g. Schmandt et al., 2002; Lumsden & Brewster, 2003; Brewster et al., 2003; Pastel & Skalsky, 2004). PDA's augmented with Touch sensitive screens can also interpret finger gestures or strokes as input, and provide audio output to the user to support eyes-free interactions with mobile devices.

Wearable computing. Wearable devices allow users to interact within smart room environments using various devices that are carried or worn by the user. This interaction provides users with persistent access and flexible control of devices through gestures (e.g. Gandy et al., 2000; Amento et al., 2002), provided that the devices are small and unobtrusive enough to wear or carry around.

2.3.9 Telematics

Computer technology is now ubiquitous within automotive design, but gestures have not yet received a great deal of attention in the research literature. Alpern & Minardo (2003) explored the use of gesturing with telematics to enable secondary task interactions to reduce the distraction caused to the primary task of driving. Additional literature explores gestures for telematics applications (Pickering, 2005) to minimise distraction while driving.

2.3.10 Adaptive Technology

Gestures are not the most common technique for adaptive interfaces since they require movement, and this may not be conducive to some physical disabilities, although some technology, such as the DataGlove, has been used to measure and track hand impairment (Zimmerman et al., 1987) in disabled users. Segen & Kumar (1998b) extends previous work to develop a system for wheelchair navigation using gestures, while the GesturePendant (Gandy et al., 2000) was also extended for adaptive interfaces for home emergency services, enabling control of devices for home patients with vision or physical impairment. Since gestures typically require movements, they may not be the primary choice for adaptive interactions, however some gestures can require only minimal motion, and can require reduced mobility than a mouse or keyboard for text entry or desktop computer interactions (Ward et al., 2000; Keates & Robinson, 1998). Pausch & Williams (1990) demonstrate the Tailor system to assist with users with speech impairment and Reilly (1998) who explores face, hand or arm gestures to control mouse movements and clicks for desktop applications. Sign language may also be considered adaptive technology for users with hearing impairment, however current systems are not yet capable of interpreting large vocabulary sets fluidly, and can experience tracking limitations in the computer-vision implementations (Fang et al., 2003; Bowden et al., 2003).

2.3.11 Communication Applications

Communication interfaces are those that seek to enable a more human-human style of human-computer interactions as described by Wexelblat (1994) and Quek et al. (2002). Gestures for communication interfaces are considered one of the most challenging areas of research by Eisenstein & Davis (2004), however research in this domain began in the 1980's, exploring speech and gesture for creating natural interactions with graphics on desktop or virtual reality displays (e.g. Bolt, 1980; Kobsa et al., 1986; Hauptmann, 1989; Weimer & Ganapathy, 1989). This research direction continued through the 1990's, with a primary focus on hand gestures and speech interactions (Bolt & Herranz, 1992; Koons & Sparrell, 1994; Wexelblat, 1995; Quek, 1994). A more recent use considers speech and pen gestures for supporting a more precise interaction with desktop or tablet computers (Cohen et al., 1997). There is also a body of research that focuses on taking a theoretical approach to understanding gesticulation as a method of enhancing speech interfaces. Work in this area seeks to understand how the two modes can be used to assist in better interpreting meaning (Kettebekov, 2004; Robbe, 1998; Quek et al., 2002; Schapira & Sharma, 2001).

2.3.12 Gesture Toolkits

Toolkits are an important approach to investigating gesture interactions. While there are several attempts at developing gesture interaction toolkits for a variety of applications (Dannenberg & Amon, 1989; Henry et al., 1990; Rubine, 1991), the individual differences in technology, gesture processing and application domains make it difficult to experience

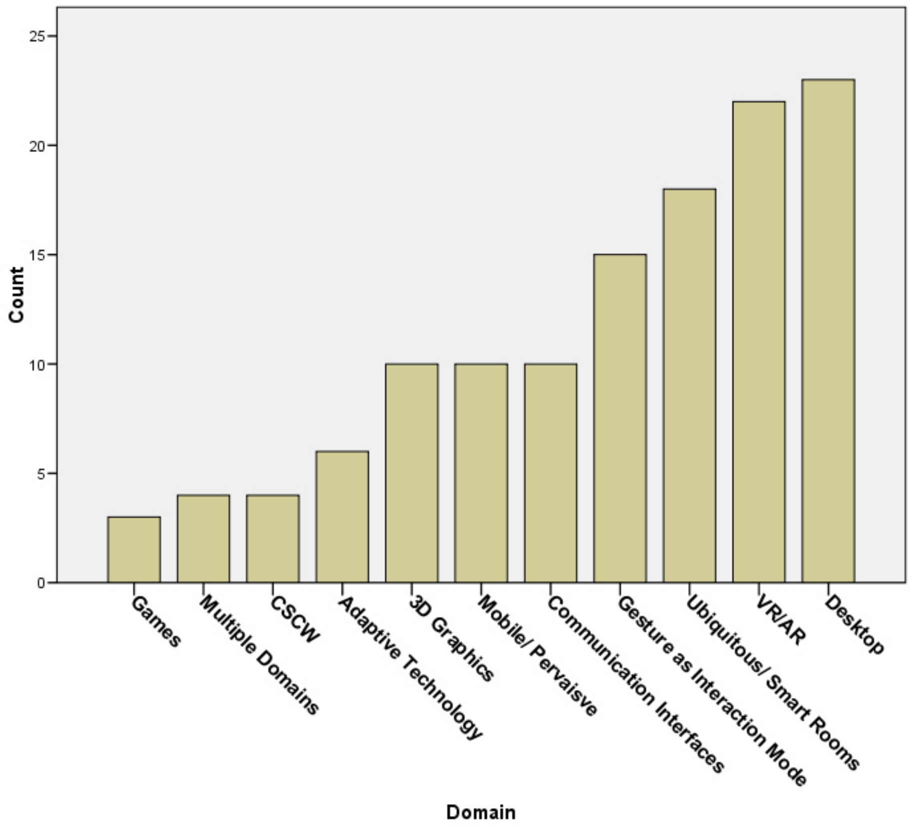

FIGURE 2.5: The different domains for which gestures are applied, according to the research literature.

any exchange of knowledge between the different interactions. But while Schiphorst et al. (2002) investigate gestures for touch sensitive fabrics, Hare et al. (2005) present a platform for developing vision-enabled gestures and Westeyn et al. (2003) presents a tool to investigate gestures enabled through multiple inputs, the research conducted in this dissertation aims to seek out a common ground between the different types of gestures within the domain of interaction research.

2.3.13 Games

Finally, we look at gestures for computer games. Freeman et al. (1996) tracked a player's hand or body position to control movement and orientation of interactive game objects such as cars, while Segen & Kumar (1998a) tracked motion to navigate a game environment. Konrad et al. (2003) used gestures to control the movement of avatars in a virtual world, Paiva et al. (2002) employed pointing gestures as a virtual-reality game input, and PlayStation2 has introduced the EyeToy, a camera that tracks hand movements for interactive games (eye, 2006). The EyeToy is currently one of the only examples we found of a readily available gesture interaction interface.

2.3.14 An Analysis of Gestures in Application Domains

While gestures are considered as an input technique for a large number of computing domains (see Figure 2.5 for a summary), there are only a few cases where they are used as standard input. One of the largest application domains for gesture interactions include virtual and augmented reality applications however, with the newer domains of pervasive and ubiquitous computing, gestures are making a prominent contribution to interactions. In ubiquitous computing, where one of the goals is to promote invisibility of devices (Weiser, 1993), gestures continue to be considered for explicit as well as implicit interactions [2] That is, an intended wave of the hand to turn on the lights is an explicit gesture, but a natural movement such as walking into a room, or sitting down on a chair can be interpreted as an implicit gesture to trigger various behaviours or actions such as temperature or lighting adjustments.

Pervasive, mobile and wearable computing domains employ gestures for eyes-free interactions that enable multitasking while reducing distraction to primary tasks such as driving or walking using touch or computer-vision interactions (Lumsden & Brewster, 2003; Pirhonen et al., 2002; Brewster et al., 2003; Pastel & Skalsky, 2004). This appears to be one of the key benefits to using vision-based gestures, along with the more natural interactions possible in the domains of teleoperations, telematics and telerobotics. However, to achieve a high level of recognition accuracy, gestures require direct input devices, which in turn can reduce the level of comfort experienced through the perceptual style interactions if computer-vision. This trade-off suggests that the technology used to enable the gestures can influence its usability. For example, mouse gestures are a commonly used interaction for web browsers, but computer vision gestures for desktop computing are not. We discuss the types of input technologies used for gestures next.

2.4 Enabling Technologies - Input

We present an overview of gesture and the technologies used to implement them. Our approach is to discuss technology from a user's perspective, focusing on the gestures they enable and some related performance measures. While there is a body of work addressing input technology and taxonomies (Buxton, 1983; Card et al., 1990, 1991; Jacob et al., 1994), our focus is on investigating technology and its affect on the gesture interactions that are possible. The rest of this chapter presents a high level breakdown and summary of technologies into perceptual and non-perceptual input to reflect the interactions they enable, shown in Figure 2.6.

[2] We make the distinction between explicit and implicit gestures based on the intent of the user. An explicit gesture is intentional, where the user consciously performs a gesture with the intent of having it recognised as such by the system, whereas an implicit gesture is a movement that the system implies to be a gesture.

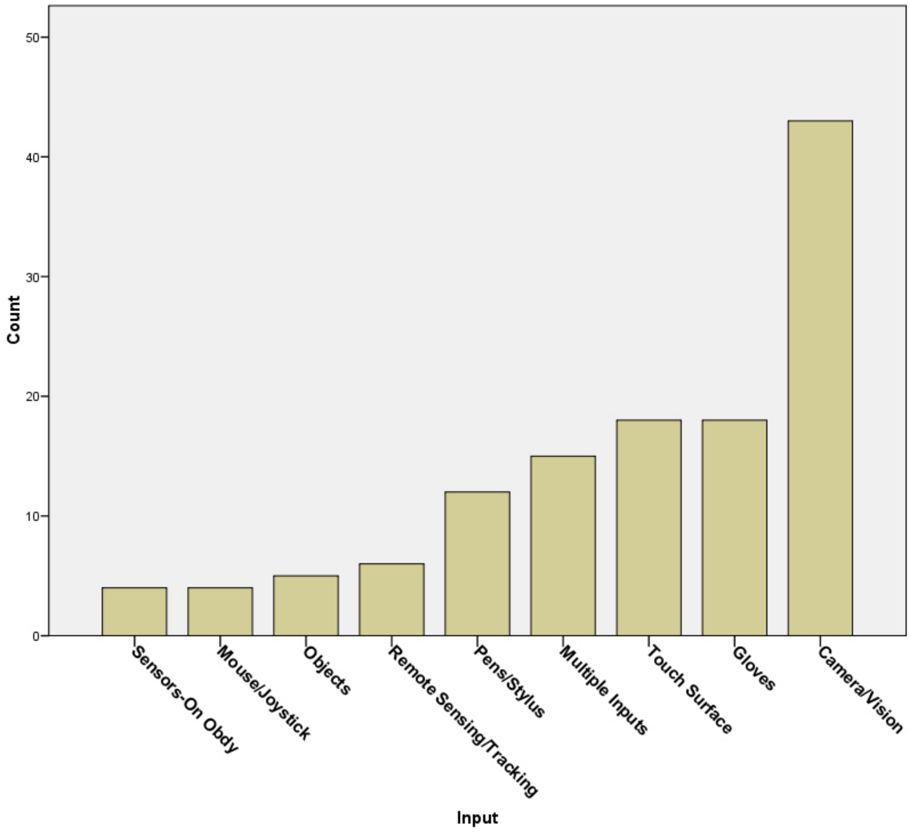

Figure 2.6: The diagram represents the distribution of the input technology used for enabling the gestures as reviewed in the research literature.

2.4.1 Non-Perceptual Input

We refer to non-perceptual input as direct input or electronic devices that require physical contact to execute a gesture. We present examples next.

Mouse and pen input. One of the first examples of a gestures was seen in 1963, when Sutherland presented his PhD work on SketchPad (Myers, 1998; Sutherland, 1963; Wikipedia, 2006b), A light pen (a predecessor to the mouse) controlled graphical on-screen objects, and like mouse gestures, enabled strokes to be interpreted as command input. Stroke gestures also take pen input (Chatty & Lecoanet, 1996) to create shortcut commands and is one of the oldest and commonly used gesture styles (Cohen et al., 1997; Forsberg et al., 1998; Rubine, 1992; Buxton et al., 1983; Moyle & Cockburn, 2002; Barrientos & Canny, 2002; Wolf & Rhyne, 1993; Jin et al., 2004; Ou et al., 2003).

Touch and pressure input. Touch and pressure sensitive screens can enable similar gestures to the mouse and pen interactions but have the added benefits of supporting interactions using fingers or hands Buxton et al. (1985); Pastel & Skalsky (2004); Allan

Christian Long et al. (1999); Gutwin & Penner (2002); Zeleznik & Forsberg (1999), which are used to enable eyes-free gestures for mobile interactions (Pirhonen et al., 2002; Brewster et al., 2003; Schmandt et al., 2002). More recently, touch and pressure sensitive materials have been considered for tabletop gesture interactions (Rekimoto, 2002; Wu & Balakrishnan, 2003; Rekimoto et al., 2003; Schiphorst et al., 2002). The addition of touch sensors can support gestures for variety of computer devices, ranging from controlling desktop monitors (Minsky, 1984), to small mobile screens (Brewster et al., 2003), to large interactive surfaces (Smith & schraefel, 2004). The gestures typically involve strokes or hand motions that manipulate objects or execute commands. Pressure can also be interpreted though weight, as in the Drift Table, where the different levels of pressure exerted by the varying weights of objects placed on the table may be considered a gesture Gaver et al. (2004).

Electronic sensing - wearable or body mounted. Bolt (1980) used electronic sensors as one of the first methods for recognising hand and arm gestures, tracking space, position and orientation through magneto-electro sensors. Space Sensing Cubes were attached to the user's wrist, tracking the x,y,z coordinates of the arm as it moved through space to track a pointing gesture for large screen interactions. These Polhemus sensors remain one of the primary devices for enabling gestures that rely on body, arm or finger movements (Bolt, 1980; Roy et al., 1994; Osawa et al., 2000; Joseph J. LaViola et al., 2001; Wexelblat, 1995). More recently Roy et al. (1994) and Osawa et al. (2000) used sensors for tracking movements in adaptive interfaces, and for navigating through virtual environments. Sensors are also used in wearable devices such as head tracking sets (Amento et al., 2002; Song et al., 2000; Brewster et al., 2003) however, these are cumbersome to wear, expensive to acquire, and difficult to implement for everyday use. There are numerous other types of wireless devices for tracking audio or visual input which are less cumbersome (Amento et al., 2002; Gandy et al., 2000), but that still require contact to be made with the device in order to gesture.

Electronic sensing: Gloves. There are several manufacturers of these devices including the Z-Glove and the DataGlove, as discussed by Zimmerman et al. (1987), and were some of the earliest methods for implementing gestures. These gloves enabled detailed detection of individual finger and hand movements for more complex gestures. The Z-Glove consisted of a cotton glove fitted with sensors that could measure finger bending, position and orientation, and that used a vibrating mechanism to provide tactile feedback. Zimmerman's system demonstrated hand gestures to manipulate computer-generated objects in virtual reality applications, and could interpret finger-spelling, evaluate hand impairment, and interface with a visual programming languages. These interactions were usually accompanied by speech, laying the groundwork for multimodal speech and gesture interfaces. The gloves also enabled interactions with graphical objects using head mounted displays for space research (Fisher et al., 1987) and for manipulating virtual object (Sturman et al., 1989; Weimer & Ganapathy, 1989). In the 1990's, glove-based gestures gained significant attention in the literature for immersed VR and autonomous agent control interfaces (e.g. Osawa et al., 2000; Song et al., 2000; Maes et al., 1997; Pierce & Pausch, 2002), telematics robotics (Fisher et al., 1987; Goza et al., 2004; Silva & Arriaga, 2003) and virtual-world navigation tasks (Zimmerman et al., 1987). For a more detailed description of the characteristics of gloves and their uses, we refer to the survey on glove-based interactions by Sturman & Zeltzer (1994).

Electronic sensing: Sensor-embedded objects and tangible interfaces. The manipulation of sensor-fitted objects are also considered as gestures in the literature (Hinckley et al., 1998). Fitzmaurice et al. (1995) demonstrated Bricks, a tangible, graspable interface where gestures are interpreted through the physical manipulation of these objects. Additional research also considers the manipulation of objects for gesture input (Paiva et al., 2002; Sinclair & Martinez, 2002; Patel et al., 2004; Wilson & Shafer, 2003).

Electronic Sensing: Tracking devices. Gestures are also executed using infrared tracking devices to detect input. Borchers (1997) demonstrated an infrared beam-emitting baton for controlling midi output. The infrared beam enables more accurate tracking by a camera, leading to more reliable gesture recognition. Additional applications for tracking gestures through infrared beams include smart room interactions to identify and control appliances or other devices (Wilson & Shafer, 2003; Swindells et al., 2002) for gestures that perform similar functions to a remote control.

2.4.2 Audio Input

An alternative method of sensing gestures is to track audio input to detect the location of a tap on a semi-public display (Paradiso, 2003). Audio sensors are not commonly applied to gesture input however this example demonstrates the use of audio perception as an alternative to vision or touch. Another application for audio sensors registers sounds made during finger and hand movements as input to a wearable device (Amento et al., 2002). Although audio is a perceptual input technology, these examples detect sounds resulting from making physical contact with the device. We next discuss perceptual input.

2.4.3 Perceptual Input

Perceptual input can support gestures that do not require making physical contact with any electronic devices such as gloves or a mouse. We refer to audio, visual and remote-sensing input as perceptual input, with the requirement that no electronic devices are required for the interaction. We present examples of perceptual style input next.

Computer Vision. Computer-vision is a major technology for enabling gestures. One of the first examples was VideoPlace (Krueger et al., 1985). Krueger used a projector to overlay a user's image onto a wall display, and to track their movements for input as they interacted with objects on the display. This technique of superimposing the user's image on top of the display was also used in the FaceSpace system (Stotts et al., 2004) where the overlay provided a form of feedback for desktop computer application. Since computer vision is not a direct-input technique, it does not support a high level or accuracy or reliability when tracking complicated movements or objects. Passive objects which are brightly coloured can be used to improve vision tracking (Hare et al., 2005; Fang et al., 2003) however, this the problem of tracking is one that is addressed in computer-vision research. Despite the recognition problems with computer vision, researchers continue to investigate vision-enabled gesture interactions, and in our research, we approach these problems from the perspective of gaining an understanding of their effect on users during interactions (see Chapter 4.

Remote sensors. Zimmerman et al. (1995) presented a variety of remote sensing devices that enabled the transmission of electric fields through a room for interpretation as gestures. Signals were shunted to a ground through a human body and an external electric field transmitted to stationary receivers to enable gestures. This was used to detect human presence and movement to enable full body, implicit gestures. Allport et al. (1995) also used electronic sensors placed on a monitor to detect finger movements and locations on a visual display.

2.4.4 Multimodal Input

The term multimodal interactions refers to two different concepts in the computing literature: one considers using multiple interaction modes sequentially Nickel & Stiefelhagen (2003), and the other considers multiple modes used in parallel such as in the combined use of speech and gesture as input (e.g. Hauptmann, 1989; Cohen et al., 1997; Gandy et al., 2000; Schapira & Sharma, 2001).

2.4.5 An Analysis of Gesture Enabling Technology

While we observed a trend towards more perceptual style input for gestures, primarily through computer-vision, the technology is not yet effective enough for use in everyday computing interactions. To the contrary, direct input devices such as pens, mice or touch interactions are already commonly used for enabling gestures in many applications. We attempt to understand this difference in uptake based on considering several factors. First, we suggest that gestures enabled through direct input devices, such as a mouse or a touch screen, can provide enhancements to interactions with devices by adding flexibility to their use through the gestures. Since these devices are most commonly used, users can more quickly learn to use these devices to perform gestures without having to first become familiar with a novel input device. Second, these familiar, direct-input devices are accurate and precise, enabling reliable and potentially complicated gesture interactions. Third, while users are familiar with vision input, most have had little or no experience with gesture recognition interactions. In addition, the overhead involved in setting up a gesture recognition system using computer-vision may not be worth the effort considering the level of accuracy that is currently possible. We begin to address some of these issues, including how accuracy rates, system speed, gesture sets and other interaction constraints can affect the user. In addition, we begin to explore gestures in terms of appropriate and relevant usage scenarios, tasks, applications, and their relationship to user requirements beginning with the study conducted in Chapter 3. We next discuss our final category for classifying gestures: system response.

2.5 System Response - Output

One of the key features we claim distinguishes gesture interactions is the different system responses or output produced using gestures. In our review of the literature, we noted that gestures, as with any input, results in an output. The result of a gesture is presented in some form to the user, and represents a major factor in the user experience. For our literature review, we approach the modality of a system response in terms of the resulting audio or visual (2D and 3D) display output, or as simply the resulting processor command (CPU). We next present system responses discussed in the literature, summarised in Figure 2.7.

Visual Output. Most system responses for gestures result in a visual display output, however this is mainly controlled based on the nature of the interaction. For example, manipulations of graphical objects on a visual display would provide persistent results of transformation on that same display. However some gesture interactions do not require visual feedback but rather, produce audio output or other types of commands for which users may not require a visual response. Visual output in on 2D displays are

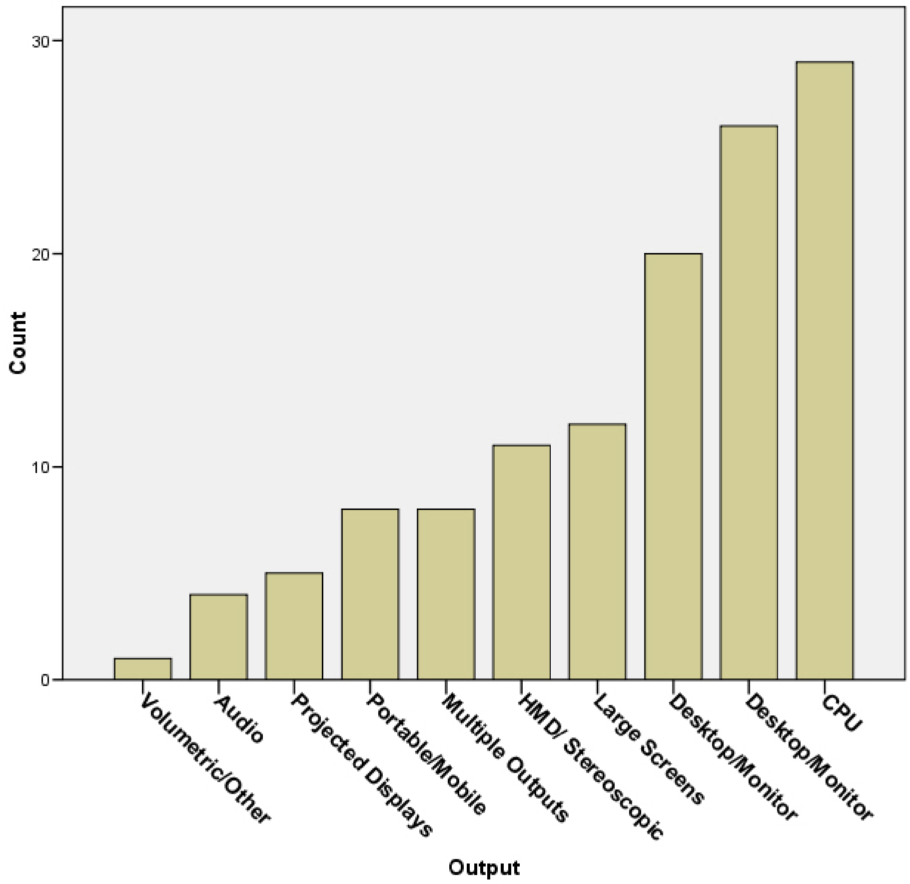

FIGURE 2.7: The bar graphs shows the distribution of system responses through output technology used in the gesture research reviewed in this paper.

most commonly related to visual interactions with desktop screens, large screen displays, projected displays and portable or mobile devices including phones and PDAs. While output is often implied by the scenario —on-screen interactions result in on-screen responses —unexpected responses are also common in gesture interactions, where an on-screen interaction may result in audio output (Schmandt et al., 2002; Pirhonen et al., 2002), or simply invoke CPU behaviour in other devices (Cao & Balakrishnan, 2003). Similarly, 3D visual output using head mounted or stereoscopic displays produce output to communicate the effects of an intended gesture to the user through the appropriate perceptual channel. The main point we wish to make in this category is that while input device can change, output devices can also change, and each must be considered individually to fully understand the nature of the interaction. For example, within virtual reality applications, the display can be immersed, semi-immersed or non-immersed, each with specific properties that can influence how we gesture. For example, fully-immersed displays may provide affordances for 3D gestures using glove input Nishino et al. (1997); Song et al. (2000), while non-immersed 3D projected displays may be more conducive to bare-hand gestures Sharma et al. (1996). 3D graphics can be presented on 2D displays (Wexelblat, 1994; Nishino et al., 1997) where glove-based manipulative gestures seem natural, and mouse gestures may not. Other visual output such as the volumetric display (Grossman et al., 2004) and 3D projection systems (Sharma et al., 1996) may also afford hand gesture interactions, however we maintain that understanding the output technology will better lead the design of suitable gestures.

Audio output. Research on gestures that lead to audio output demonstrates how a simple task can be executed and provide responses without requiring visual attention (e.g. Schmandt et al., 2002; Pirhonen et al., 2002; Brewster et al., 2003). While most audio output used in gesture research is situated around mobile and pervasive computing, the main function for non-speech audio is feedback, which is applicable to most computing domains. We investigate different types of audio response or feedback used with gesture interactions in Chapter 6.

CPU: command directed responses. Often, the response of a recognised gesture is not required to be directed to a specific output device. We refer to this as a CPU response, where the resulting output of a gesture is simply stored, or redirected to another device. For example, pointing gestures that identify devices in a smart room may lead to shutting down a device, or changing the state of the device, and may not lead to additional system responses Wilson & Shafer (2003). The responses can also lead to multiple responses as determined by the system (e.g. Pausch & Williams, 1990; Roy et al., 1994; Keates & Robinson, 1998; Reilly, 1998), or can lead to a variety of responses from different devices within smart room or ubiquitous environments (e.g. Gandy et al., 2000; Fails & Olsen, 2002; Wilson & Shafer, 2003; Nickel & Stiefelhagen, 2003).

2.5.1 Analysis of System Responses

The primary response of a gesture interaction remains directed to desktop style, 2D visual displays. However, as ubiquitous computing domains are explored, we should see a move away from the more traditional output, towards more audio and CPU responses. One trend we note is that novel output technologies often become targets of gesture research. There is a need to move away from direct input devices to seek more natural and novel interactions, and gestures can potentially fill that gap. Volumetric displays (Grossman et al., 2004) and 3D projections (Sharma et al., 1996) present opportunities for employing manipulative gestures to control the display output. However, the relationship between these novel outputs and gestures are tightly bound by the nature of the task and the enabling technologies that are associated with a given interactive system. Often these novel technologies are on the cutting edge, where accessing expensive devices make it difficult to conduct significant user studies. We discuss the different types of evaluations conducted on gesture research next.

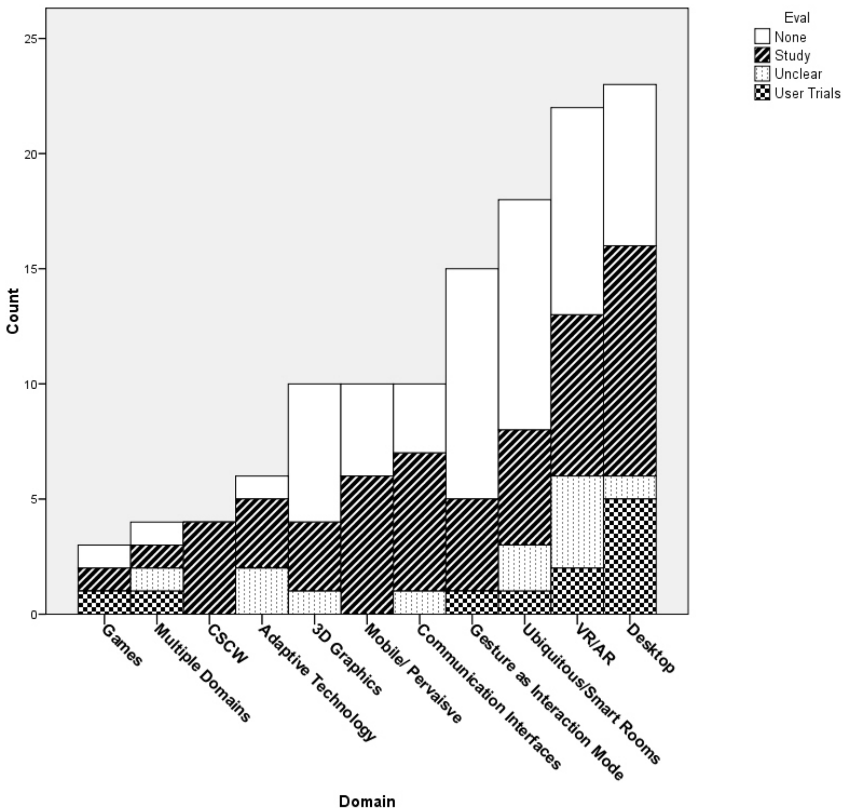

FIGURE 2.8: Evaluations of gesture research within Application Domains.

2.6 Evaluations

In this section, we discuss evaluations conducted on gesture systems, and the various problems and motivations that they have attempt to solve. Figure 2.8 shows that almost half of the systems, interactions, applications and devices that we reviewed do not include any form of system evaluation or experimentation. While this was identified as a problem with the existing research, we next discuss potential reasons for this.

2.6.1 Point Designs

Many of the systems that we reviewed for this work present point designs, or systems that either implement or propose novel interaction techniques or applications for gestures (Hinckley et al., 1998; Turk, 2004). However, given the nature of these point designs, extensive evaluations are often not conducted, providing a possible explanation into why they remain point designs, and rarely contribute empirical results to increase our knowledge about gesture interactions. Some examples of point designs are novel applications such as the shaking-gestures used as a security measure for mobile computing in public displays (Patel et al., 2004), or applications that utilise infrared pointers

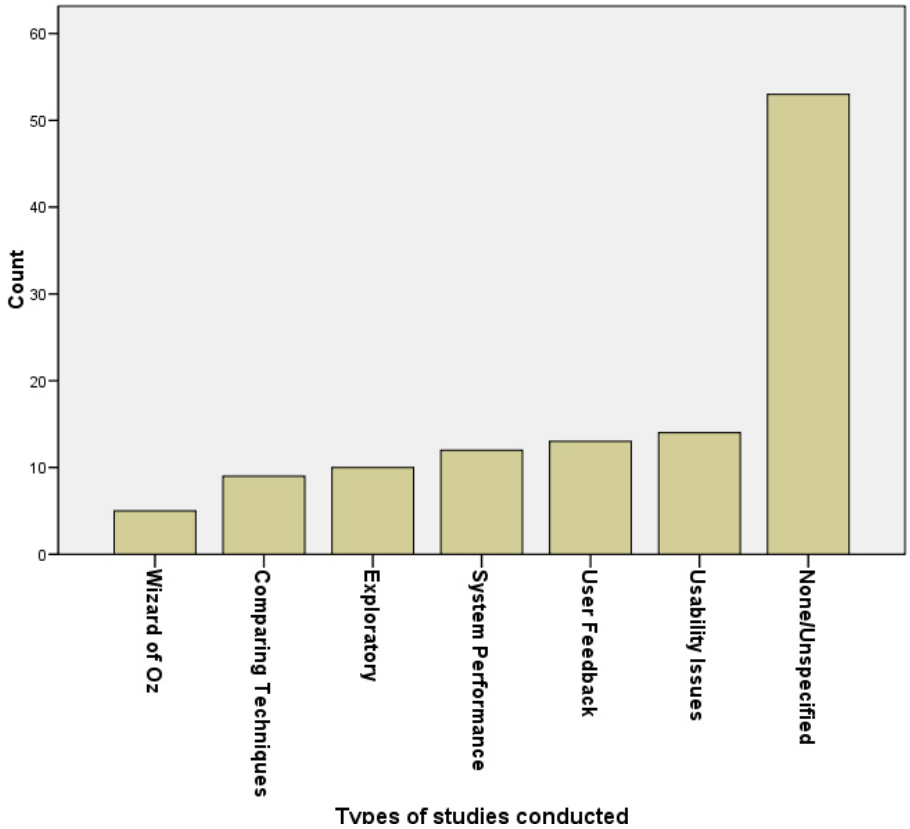

Figure 2.9: A summary of the different forms of evaluations we reviewed in the literature.

(Swindells et al., 2002), touch interactions (Rekimoto, 1997) or the physical bumping of devices to initiate a file transfer (Hinckley, 2003). While there are many examples of point designs in gesture research, we suggest that this may be one of the problems that could contribute to our understanding of why after over 25 years, work like (Bolt, 1980) has still not experienced uptake into everyday computing.

2.6.2 Evaluating Gestures

We noted several highly prominent issues that could lead to a better understanding of gestures. For example, the graph 2.11 shows almost half of the research reviewed in this paper has not presented any form of evaluation or study that is relevant to their systems or applications. Since all of the literature included in this pie refer to implemented systems, applications or interfaces, it is surprising that most of them have not performed any form of evaluation or provided any results about the affects of usability in terms of accuracy or any other features of their system. Although the research presents novel work, it would seem that there should be some form of study to contribute knowledge towards advancing gesture research. In the next section, we discuss some approaches that were used to study gestures (see Figure 2.9).

Task analysis. Although task analysis is a common approach to designing systems in HCI, there is still limited understanding of how to determine which tasks are most appropriate and suitable for gesture interactions. For example, there are many gesture systems for controlling music players, but we found no studies to investigate which functions users want to control with gestures. Although there is a large body of research addressing tasks, their analysis and characterisations (Adamczyk & Bailey, 2004; McCrickard et al., 2003b; Wild et al., 2004; Czerwinski et al., 2004), there is little attention towards determining how different input modes can effect interactions. We begin to address this issue in Chapter 3, but next, we consider the scenarios in which these tasks are situated.

Computing scenarios. Our review of the literature suggests that most interaction scenarios are designed to accommodate what the technology can handle, leading to systems that require the user to adapt to the technology, rather than learning what the user actually requires from the technology. Figure 2.10 presents some of the main motivations for exploring gestures, which include creating more natural, intuitive, or simple interactions. However, these assumptions are not always based on empirical evidence, as we did not find research that specifically addresses ways to determine if a scenario can benefit through the use of gestures. We present several studies within this dissertation that begin to demonstrate how these scenarios can be determined and evaluated, in Chapters 3 and 4.

System performance and user tolerance. Again, there are few if any studies noted in this literature review that investigate system performance characteristics and their effect on user tolerance. Some evaluations attempt to determine accuracy rates of systems, but researchers typically conduct only short user trials as shown in Figure 2.9. Results from these studies only present results that are relevant to individual systems and do not contribute to advances in the field. We begin to address the issue by investigating user tolerance levels for different system performance issues in gesture recognition in Chapter 4.

2.6.3 Analysis of Evaluations in Gesture Research

As discussed in the previous sections, we observe several trends that persistently motivate the use of gestures to create natural, simple, intuitive, human-to-human style interfaces and interactions. We also noted that a motivation for developing novel interactions whenever new technology or new application domains are introduced (Cao & Balakrishnan, 2003; Paradiso, 2003; Wilson & Shafer, 2003; Lenman et al., 2002a; Fails & Olsen, 2002; Swindells et al., 2002). However, before addressing some of the performance issues involved in these new technologies, or considering appropriate scenarios

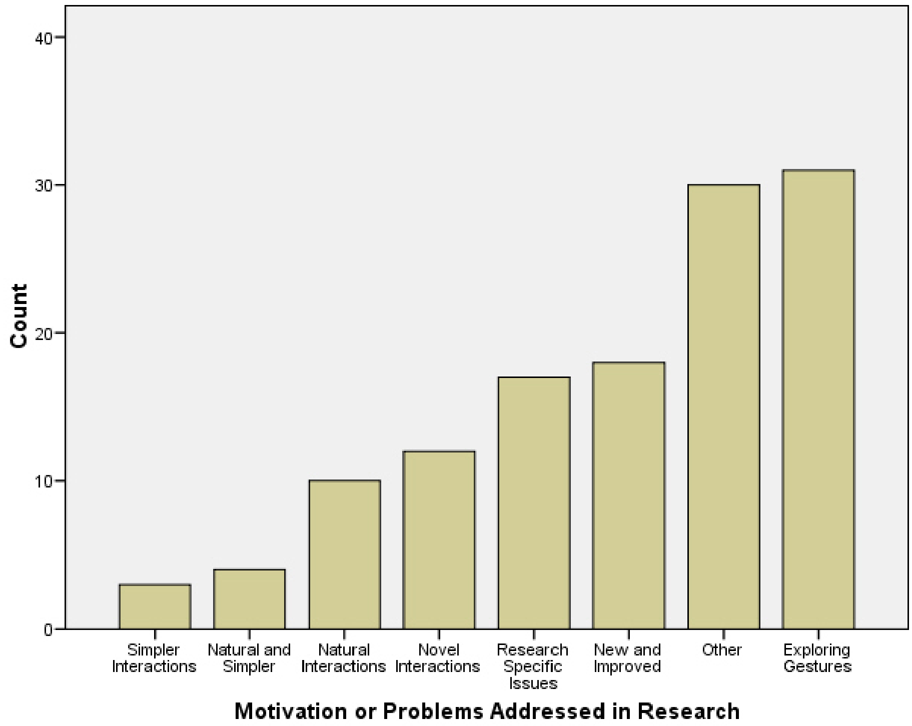

FIGURE 2.10: Motivations and problems addressed through the gesture research.

for application domains, we propose that existing research may contain the information that can assist in determining many of the functions and constraints of interactions if we could understand the relationships that exist between the technology and the humans. This is the intended application of our classification, and our framework, presented in Chapter 5.

2.7 Gestures: Addressing and Revealing Problems

We examine some of the motivations for considering gestures as an interaction technique, summarised in Figure 2.10. The graph shows that along with solving specific problems and generally exploring gestures, the main motivations are to create more natural, novel and improved interactions. However, gestures also create problems due to their implementation, which we discuss next.

Natural interactions. One of the major motivations for using gestures is the creation of natural computer interactions. Gesticulation is referred to as a natural gesture, however researchers are a long way away from understanding how to interpret gesticulation in the context of speech before such a system can be implemented (e.g. Wexelblat, 1994; Quek et al., 2002; Eisenstein & Davis, 2004; Kettebekov, 2004; Koons & Sparrell, 1994). Gestures are also proposed for enabling more natural interactions using the DataGlove (Fisher et al., 1987; Bolt & Herranz, 1992; Zimmerman et al., 1987), however, there is a trade-off to make between accuracy and naturalness. Alternatively, perceptual gestures that can free the user from devices also suffer from inaccurate recognition and a lack of control that may be more frustrating than useful.

Simplifying interactions. The call for simpler and more intuitive interactions with computers through coverbal or multimodal speech and gesture interfaces has dominated the literature since at least the 80's (e.g. Bolt, 1980; Kobsa et al., 1986; Hauptmann, 1989; Weimer & Ganapathy, 1989; Bolt & Herranz, 1992; Koons & Sparrell, 1994; Quek, 1994; Wexelblat, 1994; Cohen et al., 1997; Gandy et al., 2000; Quek et al., 2002; Eisenstein & Davis, 2004; Kopp et al., 2004), combining speech and gestures as a means of creating a human-to-human approach to computing. But while this approach assumes that human-to-human interactions, when applied to a computer would be useful, there is little evidence to support this. Stroke and mouse gestures do however demonstrate examples of gestures creating more simplified interactions, enabling hand writing to be interpreted as input, and mouse gestures for shortcuts to menu access.

General improvements for interactions. A general problem that motivates gesture research is its potential to improve interactions. Gestures can improve interactions by enabling meaningful pen strokes for drawing and for introducing quick commands to control applications (Buxton et al., 1983; Rubine, 1992; Cohen et al., 1997). Gestures for 3D graphic interactions enable additional degrees of freedom over the mouse, by using hand gestures to control virtual or real objects (Segen & Kumar, 1998a; Zeleznik & Forsberg, 1999), and to support creative, lightweight interactions for smart room environments (Streitz et al., 1999; Gandy et al., 2000; Rekimoto, 1997). Adaptive interfaces also demonstrate some potential improvements where gestures can control wheel chairs or enable text input without having to use a keyboard (Pausch & Williams, 1990; Keates & Robinson, 1998; Ward et al., 2000). Additional improvements are suggested in pervasive and mobile computing domains (Amento et al., 2002) and gaming applications (Freeman et al., 1996), creating intuitive interactions based on using real-world objects.
.

2.8 Classifying gesture interactions

We now present a the four categories we determined as representative candidates for classifying gesture-based systems and interactions. Each category was selected out of the complete list derived from our review of the literature list (see Appendix A). These

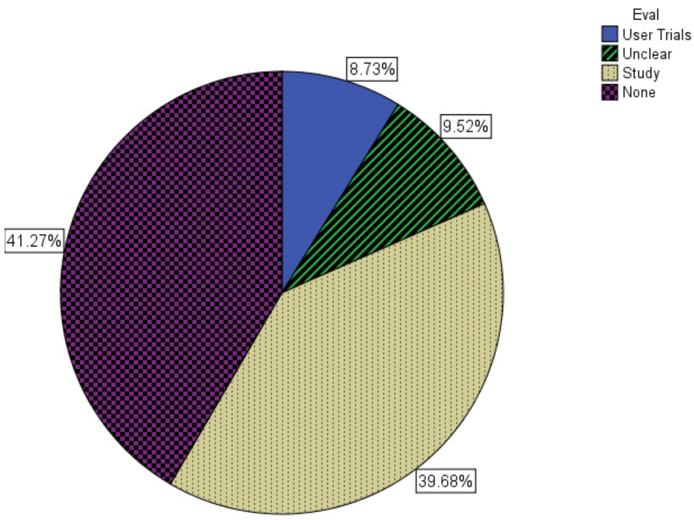

FIGURE 2.11: Represents the distribution of the studies conducted in the reviewed literature.

categories are intended to provide a fundamental understanding of gesture interactions across different technologies, applications, and techniques. We selected gesture style to address the physical nature of the interaction, application domain to provide contextual information about the general nature of the interaction, input (enabling technology), and output (system response) to represent the tangible or hardware components of gesture interaction systems,

Category 1: Gesture Styles. We argue that when considering any gesture system, it is important to identify the type of gesture intended for the interaction. While other factors such as the type of input technology or the intended tasks can influence the choice of gesture, it is necessary to acknowledge the role of gesture style in the interaction design. Although many researchers often combine styles, there remains a fundamental difference required for the processing of each. For example, while semaphoric gestures can be programmed into the recognition system, and identified in real time, a pointing gesture would require more information about the position and spatial location of the user within environment. However each style can be addressed to inform the processing requirements for the system. A summary of the five gesture styles described in this review —gesticulation, manipulation, semaphores, deictic and language gestures —is presented in Appendix A.

Category 2: Application Domains. Application domains provide us with a structure for understanding tasks and their characteristics within specific interaction contexts. For example, desktop interactions can be understood in terms of the basic setup of a computer, a monitor, and input devices (typically a keyboard and a mouse), constrained to an individual work space. A virtual reality context may be less definitive in terms of the input and output devices and their proximity during interactions. The inclusion of application domain as a category provides a level of understanding about the input and output devices, and the tasks that are well understood by the practitioners within that domain. Examples of the different application domains reviewed for this research were presented earlier in this Chapter in Section 2.3.

Category 3: Enabling Technology (Input). We refer to the different types of input that can enable gestures as enabling technology for the purpose of this classification. The inclusion of enabling technology is essential for gesture classification, since all gesture require some device to enable its recognition. We can refer to enabling technologies in terms of the interaction characteristics that affect the user experience (response speed, error rates, input constraints). We can also express these functional characteristics in relation to user satisfaction when we map performance measures onto user experiences. Though this is well understood area of computer interactions, its inclusion in this classification plays the major role in determining the type of interaction that will be possible with gestures, and what the interaction will be like.

Category 4: System Response (Output). We refer to the different types of output that can result from a gesture as system response, since gestures may not always produce output that is intended for user consumption. System response represents the hardware for which the intended action of a gesture is directed. It is included to provide a complete picture of the interaction, both in terms of the modality of the output, and the nature of the interaction.

2.9 Summary

We presented a literature review of gestures as an interaction technique, organised using the concepts that we identified and coded using content analysis techniques. Continual analysis of the categories through our accumulation of data from the literature informed our classification of gesture interaction systems into four categories that represent components if gesture interactions: gesture style, application domain, enabling technology, and system response. Over 40 years of computer research in gestures suggest they are may provide a natural, novel and improved interaction technique, and that while perceptual input is less reliable than direct input devices, gestures maintain their presence in the research. Unlike speech interactions, which are included as standard features of Microsoft Windows or Mac operating systems, we ask why not gestures? This was a problem discussed by Buxton et al. (1983) who noted that for gesture interactions, there exists

> a perceived discrepancy between the apparent power of the approach and its extremely low utilisation in current practice.

This is also relevant today, where so much is done in theory, yet so little is ever applied. In the next section, we present a study which begins to uncover some of the defining characteristics of the categories presented in this chapter.

Chapter 3

Investigating Functional Utility of Gestures

"Our Age of Anxiety is, in great part, the result of trying to do today's jobs with yesterday's tools."

Marshall McLuhan

3.1 Introduction

An initial motivation for our research was the iGesture system, an extensible platform for conducting gesture research, developed at the University of Southampton Hare et al. (2005). For several months, we studied the iGesture system, measuring its performance, gesture recognition capacity, and functionality. Results were generally positive, however our knowledge was limited to the one system. If we were to consider making changes to improve that system, we would first have to understand what we wanted to improve and how we could improve it. To do this, we stepped away from the technology, and considered the human perspective to learn if gestures could indeed enhance interactions. Our review of the literature revealed semaphoric gestures were common in research, yet rarely seen outside of the laboratory setting. In addition, several researchers had suggested that hand-based signs, or semaphoric gestures are not natural, and are an infrequent form of human gesture (Wexelblat, 1998; Quek et al., 2002; Kettebekov, 2004). Wexelblat (1998) in particular challenged the use of gestures as a viable interaction controller since they could be physically taxing and demonstrated at best 90% recognition accuracy. While we would not normally expect people to settle for 90% accuracy in recognition, we challenged this claim. We undertook a study to investigate gestures and their functionality using the following approach:

1. We investigated a functional utility - a value to the interaction - for semaphoric gestures

2. We considered different contexts and compared the affects of using gestures for secondary tasks.

We approached this phase of the research using the Wizard of Oz (WoZ) methodology, a term coined by J.F.Kelley (2002)[1] to describe an approach where the intelligent behaviour of a computer was controlled by a person who remained out of view of the participants. The term was inspired by the film, the Wizard of Oz, and is an efficient method for investigating interactions without having to deal with many inhibiting system constraints. We chose this method to avoid the constraints that iGesture could potentially impose on the experiment due to sensitivity to lighting changes and tracking. Our experience with iGesture and the user studies we conducted confirmed our initial thoughts that gestures would be useful for controlling music applications. This informed our approach to this study, with results suggesting that gestures offer significant benefits over function keys for secondary task interactions in multitasking situations. We described secondary tasks as being on-critical and placing little cognitive demand on users. This result supported the goals of notification system interactions, where multitasking is a key feature of the interaction and there is a goal to reduce distraction to users. This result also inspired a collaborative effort with the Virginia Tech to investigate gestures for notification system interactions, discussed in Chapter 6. The remainder of this chapter describes studies conducted on iGesture, and the experiment to determine the functionality of semaphoric gestures. A short version of the experiment appeared in the 2005 conference on computer human interactions (CHI) (Karam & schraefel, 2005a).

3.2 iGesture

We present a report of several studies to investigate the iGesture platform, which supported some of the initial research conducted towards this dissertation. iGesture played an integral part in guiding our qualitative and formative studies, and is currently being redesigned for future research and development. We next present details of the iGesture system, and the experiments we conducted to determine its performance in terms of accuracy rates, its capacity to recognise gestures, and the different types of gestures it could recognise. We also discuss some lessons learnt from our experiences working with gestures and iGesture.

[1]The term was said to have been coined circa 1980 by Kelly, which is stated in his current web site.

FIGURE 3.1: Multiple users controlling a music application using iGesture and a coloured marker to assist with tracking.

3.2.1 iGesture Platform

The iGesture platform is a tool for implementing multimodal gesture-based interactions in multimedia contexts. It is a low-cost, extensible system that uses visual recognition of movements to support gesture input. Computer vision techniques support interactions that are lightweight, with minimal constraints. The system recognises gestures executed at a distance from the camera, for multimodal interaction in a naturalistic, transparent manner for many different application domains including desktop, ubiquitous and CSCW computing environments (see Figures 3.1 and 3.2). In addition, iGesture can process raw visual input to control midi devices. For our research, we focused on semaphoric gestures for controlling software application tasks. iGesture is scriptable and can map a gesture onto any command line function or statement. While this work was exploratory, the experience with iGesture contributed to understanding of many characteristics of the gesture interactions that we investigated in this research. We investigated several gesture recognition systems in the literature, (Westeyn et al., 2003; Dannenberg & Amon, 1989; Henry et al., 1990), but none provided the flexibility ease of use of iGesture. iGesture was easily configurable, gestures were easily trained and mapped onto tasks so that we discovered many gestures that it could recognise. We discuss our observations and experience from using iGesture next.

Application controls. iGesture runs on the Mac OSx operating system, and can most software applications. For example, in iTunes, an Applescript can be written to control most tasks —play, pause, stop and volume controls, as well as more complicated tasks such as managing play lists. We interacted with other applications including Winamp, Quicktime, Microsoft Office, and a jukebox software application written by Max Wilson. A screen shot of the iGesture system is presented in Figure 3.3.

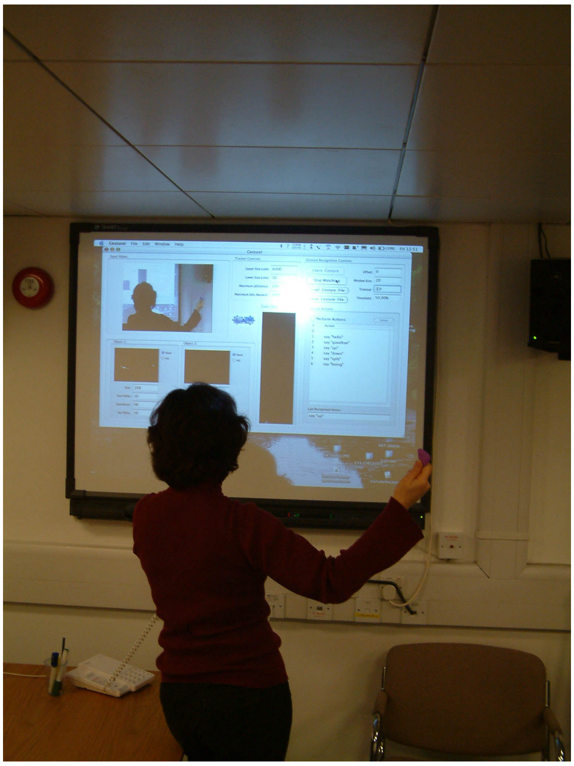

FIGURE 3.2: Training the iGesture system on a large screen display.

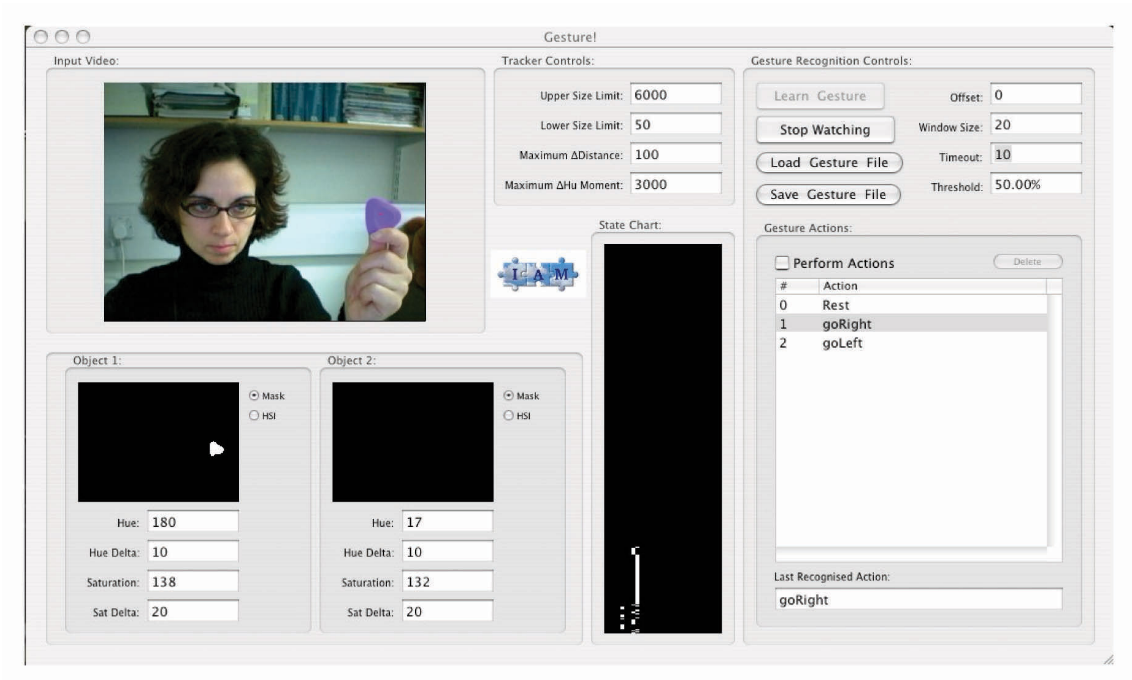

FIGURE 3.3: Screen shot of the iGesture interface.

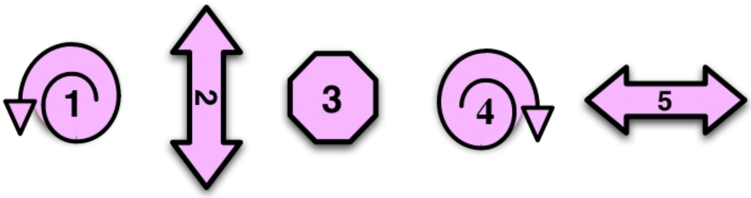

Figure 3.4: Graphical representations of the types of movements associated with different gestures presented to participants during the training sessions to provide visual cues for each gesture.

Gesture sets. iGesture the system is designed to track directional motion and could recognise combinations of horizontal, vertical, and circular motions. Figure 3.4 shows diagram of one set of gestures we tested. Gestures can be programmed using two channels, so that there are identical gestures that the system can recognised for both left and right hand interactions. We use two different coloured objects to track movements. Hand recognition was possible, however it was more effective to use a bright colour since there was less chance of similar colours being picked up accidentally in the background. We discuss recognition next.

3.2.2 System Performance

While iGesture recognises a large set of gestures, several issues lead to poor recognition. First, variations in the lighting requires the hue and saturation levels to be altered to reflect the changes in light. In addition, gestures with similar trajectories can lead to incorrect recognition, while recognition performance decreases with each gesture trained in the system. A discussion on the implementation details of the iGesture system is provided in a technical report by Hare et al. (2005), at the project web site Karam & Hare (2004) and in Appendix B of this dissertation, along with links to demonstration videos. We next discuss the studies conducted to determine iGesture accuracy rates.

3.2.3 Measuring Performance Accuracy: Experiment

To check the accuracy of the semaphores gesture recognition subsystem, we performed an evaluation in controlled conditions. The system was set up in a room with fixed lighting and the camera was positioned to cover as much area as possible. The fixed lighting conditions mimic the office environment in which the system is currently deployed. The evaluation was designed to assess the accuracy rate of the system in terms of percentage of correctly recognised gestures, the percentage of false positives (gestures incorrectly recognised) and percentage of false negatives (gestures not recognised). A gender balanced group of 8 volunteers was assembled for the evaluation in order to assess the performance over a group of potential users.

TABLE 3.1: Averaged results from each part of the evaluation

	Average Correct	Average Incorrect	Average Missed
Pre-trained, Centred	93%	4%	3%
Pre-trained, Non-Centred	84%	5%	12%
Subject-trained, Centred	91%	3%	7%
Subject-trained, Non-Centred	87%	2%	11%

Method. The evaluation was performed in two parts. First, the system was loaded with a set of pre-trained gestures, illustrated in Figure 3.4, however for this experiment, we only used single handed gestures. Second, the participants were asked to train the system to recognise their gestures before the evaluation commenced. These two parts allowed us to evaluate the effect of user-trained versus pre-trained gestures on the recognition accuracy. Both parts of the evaluation consisted of two subparts. First, the subjects were asked to perform each of the gestures 5 times in a stationary position directly in the centre of the camera's field of view. In the second sub-part, five different points in the room were pre-selected to cover the full visual field of the camera. The participants were then asked to perform the gestures at each point, while facing the camera. The results of the evaluation showed no statistically significant intra and inter-subject variability, so the results have been averaged and are shown in Table 3.1.

Results. Extensive use of the iGesture system provided us with the following details about which semaphoric gestures were best recognised by the system. Since iGesture tracks movements in 4 directions, we had to work around conflicts between gestures that follow a similar initial trajectory. We noted that right and clockwise, left and counter-clockwise, up and stop gestures were the most confusing to the system. Users also recognised this conflict becoming frustrated with them and requesting that the gestures be changed to avoid these problems. We noted that the primary researcher became proficient at all of the gestures after extended periods of use and could avoid these conflicts using various strategies. This was not possible for novice users, who did not have enough experience to create strategies for improving recognition. Because of the amount of time required for users to become proficient at performing gestures, and due to the changes in lighting throughout the day that would effect tracking, we continue to use the (WoZ) methodology to enable us to continue with our focus on the interactions and not on the system.

3.2.4 iGesture and Manipulative, Free-form Gestures

An additional feature of the interaction enabled through the iGesture system included a direct transfer of visual images to midi input to create a simulation of an air guitar interaction. Figure 3.5 shows a participant using the air guitar feature of the iGesture

FIGURE 3.5: A participant enjoying the air guitar interaction during a free-form gesture investigation.

platform. With this application, we were able to design a creative interface that was mostly used to demonstrate the more playful side of gesture interactions. While this interaction provided a great deal of enjoyment for the researchers and visitors to the University, this interaction is included to stress the flexibility interactions possible using iGesture.

3.2.5 Experience Report and Research Motivation

While extensive use of the iGesture system leads to improved performance on gestures, the system is still in its prototype phase, and is currently being redesigned for multiple platform use. We will also implement different recognition algorithms and techniques such as tracking shapes in addition to colour for future testing. We continue to use the system, discovering novel ways of performing gestures, and working around the performance issues. Other gesture recognition systems and techniques can enable more robust recognition than iGesture, and we could redesign the recognition process however before embarking on this task, we wanted to first ensure there would be a functional utility for these interactions. We decided to learn more about the human perspective of gesture interactions, and the scenarios in which gestures could be of benefit to the user. In the next section, we present research that attempts to determine if there is indeed a functional utility for semaphoric gestures, and in what contexts.

3.3 Gestures and Secondary Task Interactions

We conducted a study to assess the roll of semaphoric gestures as a secondary task interaction technique, resulting from a series of studies and experiments. We describe how we gathered qualitative information through a series of interviews and observations,

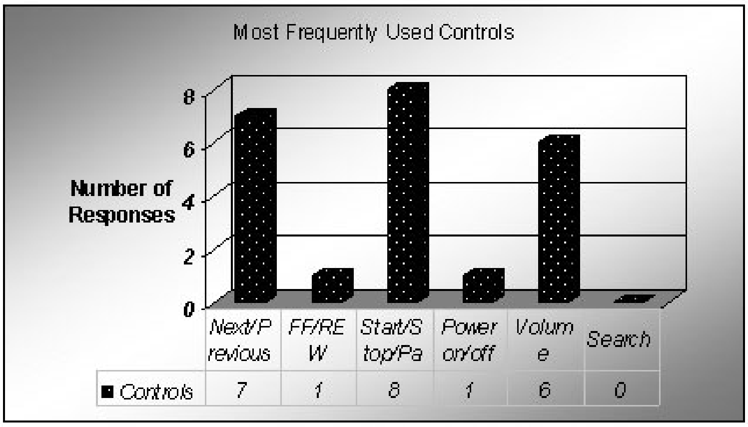

TABLE 3.2: Table shows tasks most commonly used, as determined in our interviews.

which informed our understanding of gestures and secondary task interactions. This was followed by an experiment to test our hypothesis about gestures and their value in multitasking situations for secondary task interactions.

3.3.1 Designing a Gesture Set

To approach the design of a gesture set, we conducted interviews with 25 people to determine if there was a consistent and natural set of semaphoric gestures that could be used for controlling the main functions of a music system. We asked participants to perform hand gestures for a each word in a list that we read aloud. Figure 2 summarises our interview results, showing that for all but play functions, most participants had a similar direction-based gestures in mind for each of the controls of a music player. Participants said they chose those gestures to reflect symbols commonly associated with music player controls, shown in Figure 3.6. For our gesture set, we settled on four controls - play, stop, previous and next tracks - for interacting with a music player. The gestures we used for the experiment included a clockwise circular motion for play, a left to right hand movement to signal next track, a right to left hand movement for the previous track and the palm held vertically facing away from the participant to stop the music. Participants were proficient in remembering and performing the gestures within only a few trials with each. In addition, we provided a questionnaire to our participants to learn about their music listening habits, and show that most participants listen to music as a background activity, using a computer as the primary music player. Figures 3.7 show the distribution of these responses to contexts for listening to music. These results motivated our choice of using a multitasking scenario, where controlling the music player is considered a secondary task.

FIGURE 3.6: The picture shows many music player controls and their use of arrows to indicate the different controllers.

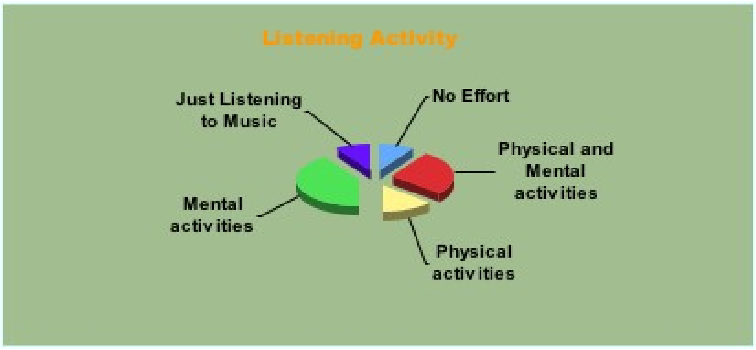

FIGURE 3.7: The graph shows that most people multitask while listening to music.

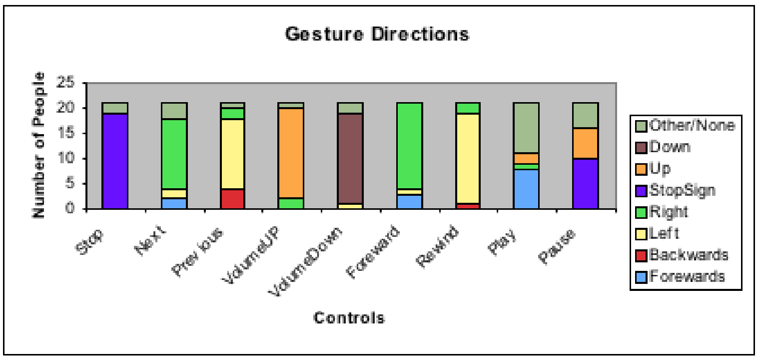

TABLE 3.3: The table presents the results from our interviews for the four music player controls used in this study.

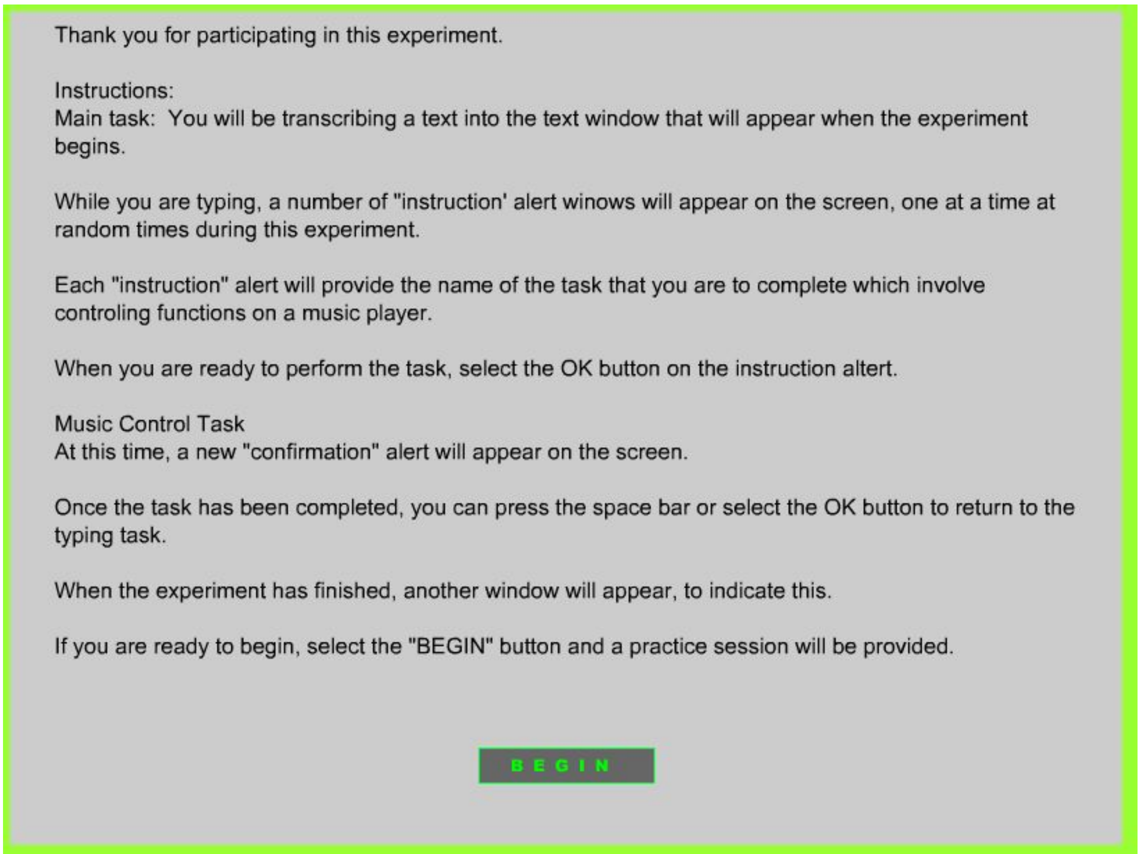

FIGURE 3.8: Screen shot of the instructions for the experiment.

3.3.2 The Experiment

Results from our interviews informed the design of our scenario, the tasks and the gestures that we hypothesised would demonstrate the benefits of gestures. The experiment was a within-participant design, and details or its design are presented next.

3.3.2.1 Tasks

Screen shots of the application that was used in the experiment are presented in Figures 3.8, 3.9, 3.10, and 3.11. We discuss the primary and secondary tasks in detail next.

Primary tasks. Participants were given a set of snap cards, each displaying either a single word or a picture (see Figure 3.12). Participants were asked to turn over a card, type the name of the object or the word from the card into the text editor window, and then to repeat this activity for the duration of the experiment.

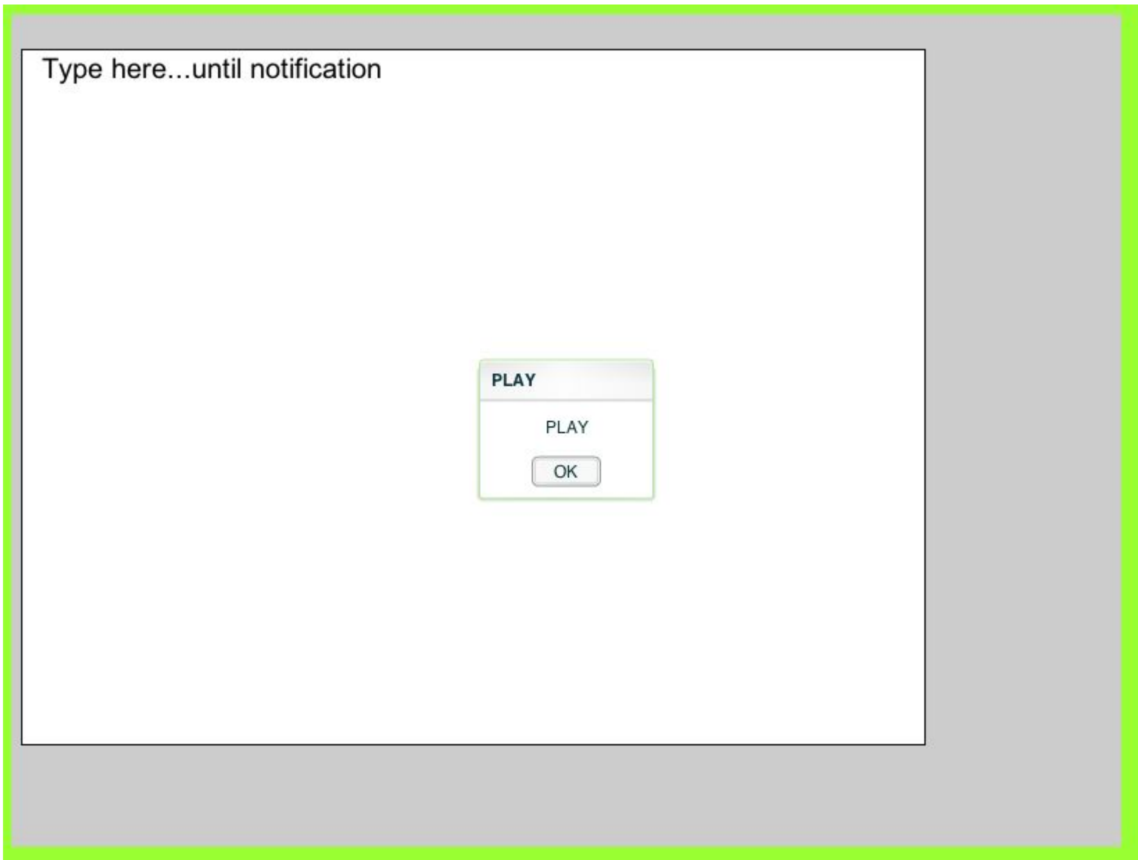

FIGURE 3.9: Screen shot of the page on which users were to perform their primary task of typing the text from the snap cards into the text window as they went through the deck. A pop-up window would be displayed at intervals throughout the experiment to alert participants to complete the secondary task displayed on the alert window.

Secondary tasks. The secondary task consisted of three steps. A click on an alert to indicate the beginning of a task, the execution of the specified control for the music player, and another click to indicate completion. This method enabled us to obtain precise timings for the duration of a secondary task through computer logs.

3.3.2.2 Experimental Design

The study evaluates a single factor —secondary task input mode —with two conditions —gestures as the experimental condition and keyboard function keys as the control condition. 8 males and 8 females completed 20 tests for each condition. The experiment was a repeated measures, mixed model design with interaction mode as the within-participant variable, and the order it was presented or counterbalancing as a between-participant measure. We looked at gender as a covariate for the analysis. Sessions lasted approximately 1 hour. Exposure to the treatments was counterbalanced in a within-participant, repeated measures design. We developed the following set of measures as a

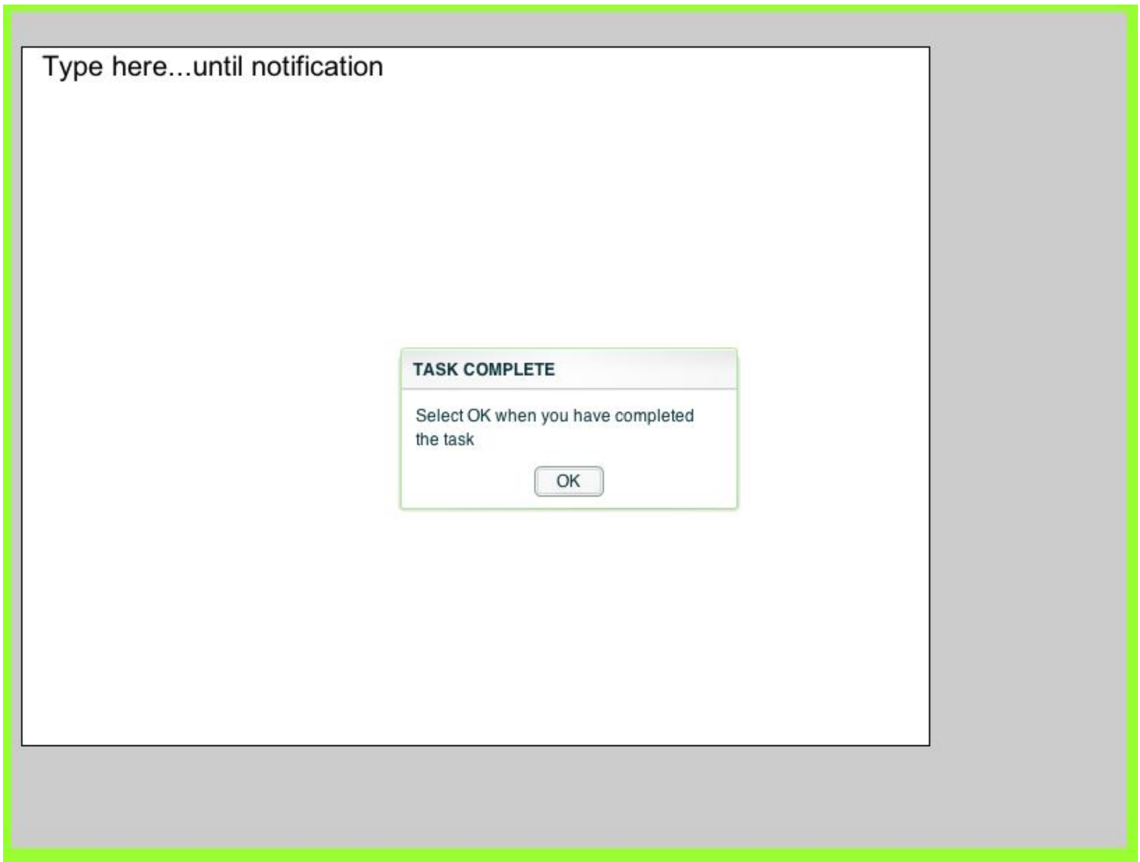

Figure 3.10: Once the initial alert is clicked, a new alert appears on screen while participants perform the secondary task.

means of evaluating the quantitative affects gestures have on the participants to complete the secondary task, producing minimum interruption to the primary task.

Apparatus. A laptop computer was used for performing the primary tasks, placed on a desk in front of the Wizard. A deck of snap cards were placed at the left of the laptop, and a keyboard was located on the right on top of a sheet of paper. During the keyboard condition, we provided an list of the function keys and their corresponding music controls. During the gesture condition, we provide a list of the gestures and their corresponding music controls (see Figure 3.12).

Variables. A notification pop-up window that contained a single word, Play, Stop, Next and Previous, corresponding to the task that the participant was to complete appeared at different intervals on the participant's screen. The participant would then perform the appropriate action. In the gesture condition, gestures were used to control the music playback; in the control condition, keyboard with function keys programmed to control the player were used. We recorded the following variables to measure user performance on the interaction:

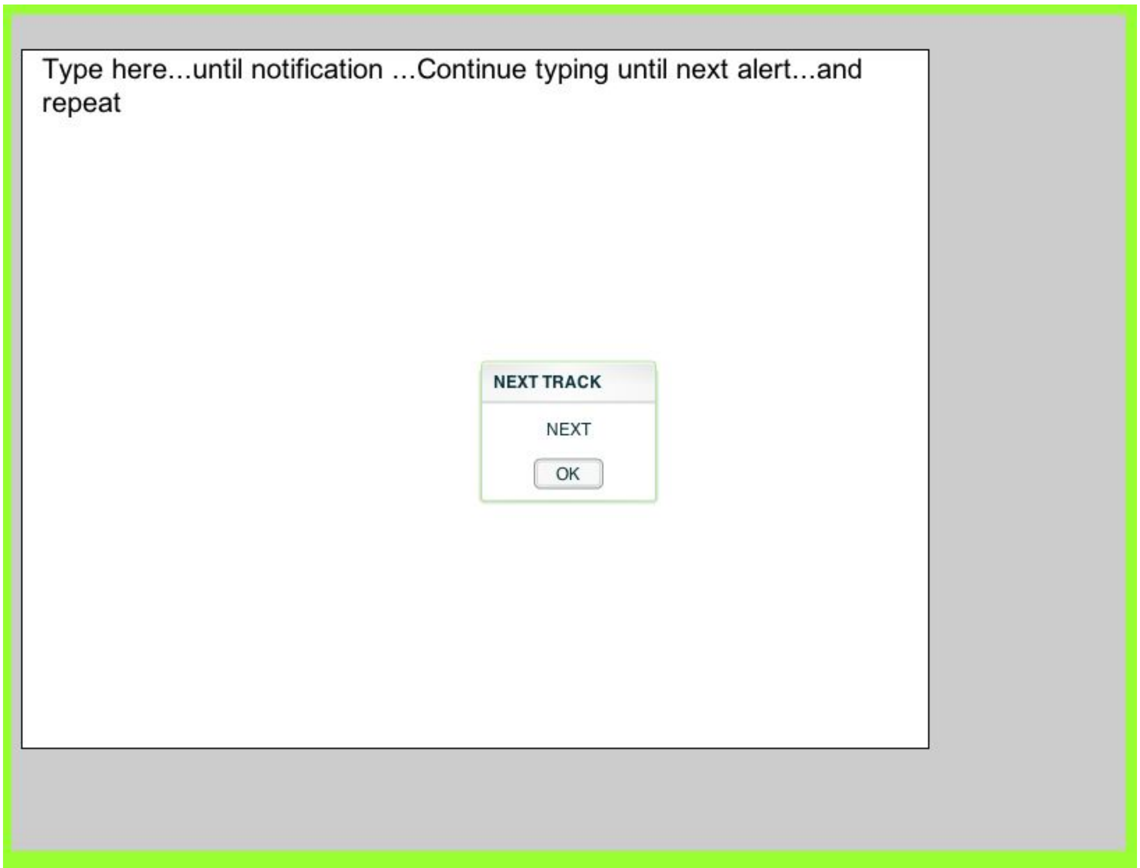

FIGURE 3.11: Once the secondary task is complete and the participants clicks on the alert shown in the previous image, they return to their typing task, until a new alert appears, and starts the cycle again.

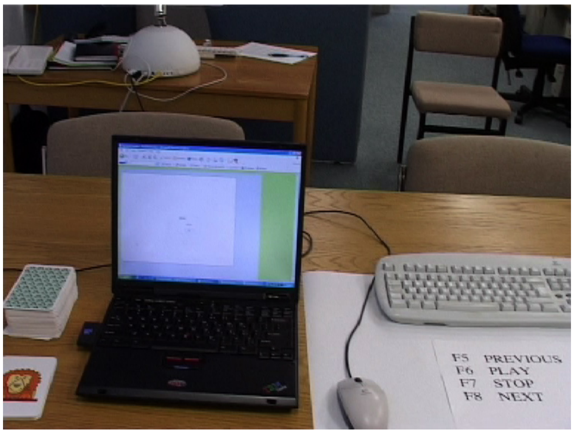

FIGURE 3.12: This photo shows the set up of this experiment. Participants were seated at the lap top and the wizard was seated across in front of the iMac computer.

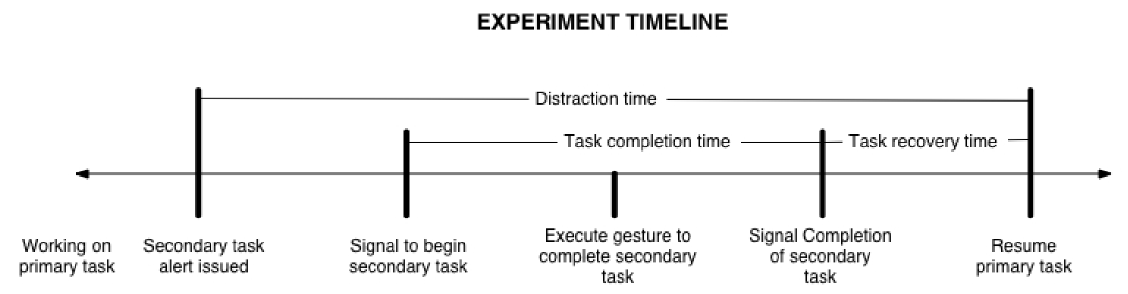

FIGURE 3.13: The timeline shows the order and alerts used to identify and measure the variables used in the experiment.

- Task Recovery Time: Represents the amount of distraction between completion of the secondary task and returning to the primary task.
- Distraction Time: Represents the total time of distraction caused to the primary task, measured from when the alert is issued to when the participant returned to the primary task.
- Task Completion Time: Represents the time taken to complete a secondary task, measured as the time between when the user signals the start of secondary task, and the completion of the task.

A timeline is presented in Figure 3.13, outlining the details of the variables and the measures that were used in the experiment.

Post experiment interviews. In the post experiment interview, we asked participants to compare gestures to the keyboard interaction. Each question asked participants to rate the gestures as equally, more or less satisfying, comfortable, preferred and distracting than the keyboard.

3.3.3 Results

We next discuss the results from our quantitative analysis, and from our observations and the subjective results. Results from the ANOVA is presented in Table 3.4, and the descriptive data is presented in Table 3.5.

3.3.3.1 Errors

While the Wizard intended to respond to all gestures without error, it is not possible to ensure 100% recognition accuracy due to the potential of human error. However, we recorded the errors made by the participants, and noted a mean rate of .02% error in gesturing, and a mean rate of .03% for the keyboard interaction across all participants.

Tests of Within-Subjects Contrasts

Source	Measure	WithinMode	Type III Sum of Squares	df	Mean Square	F	Sig.	Noncent. Parameter	Observed Power[a]
WithinMode	DistractionWithin	Linear	.066	1	.066	.003	.960	.003	.050
	TaskTimeWithin	Linear	6.057	1	6.057	.639	.425	.639	.125
	RecoveryWithin	Linear	11.478	1	11.478	5.843	.016	5.843	.674
WithinMode * Gend	DistractionWithin	Linear	.823	1	.823	.032	.859	.032	.054
	TaskTimeWithin	Linear	15.898	1	15.898	1.678	.196	1.678	.252
	RecoveryWithin	Linear	5.263	1	5.263	2.679	.103	2.679	.372
WithinMode * ModeOrder	DistractionWithin	Linear	192.611	1	192.611	7.402	.007	7.402	.774
	TaskTimeWithin	Linear	26.056	1	26.056	2.750	.098	2.750	.380
	RecoveryWithin	Linear	.205	1	.205	.104	.747	.104	.062
Error(WithinMode)	DistractionWithin	Linear	7988.741	307	26.022				
	TaskTimeWithin	Linear	2908.957	307	9.475				
	RecoveryWithin	Linear	603.065	307	1.964				

a. Computed using alpha = .05

TABLE 3.4: The table shows the results from the ANOVA conducted on the within-participant variables used in this model. The results are described and discussed in this section.

Descriptive Statistics

	Mode Order	Mean	Std. Deviation	N
Distraction Time: Gestures	Keybd First	4.67957	6.047249	155
	GestFirst	3.79927	.850471	155
	Total	4.23942	4.333621	310
Distraction Time: Keyboard	Keybd First	3.72731	.935797	155
	GestFirst	5.07555	8.158778	155
	Total	4.40143	5.836735	310
Task Time:Gestures	Keybd First	2.47762	6.031007	155
	GestFirst	1.86308	.651373	155
	Total	2.17035	4.293465	310
Task Time:Keyboard	Keybd First	1.66761	.672305	155
	GestFirst	1.95905	.877970	155
	Total	1.81333	.794189	310
Recovery Time: Gestures	Keybd First	2.38238	1.522386	155
	GestFirst	2.09568	1.133441	155
	Total	2.23903	1.347576	310
Recovery Time: Keyboard	Keybd First	2.64256	2.095416	155
	GestFirst	2.47503	1.192557	155
	Total	2.55879	1.704146	310

TABLE 3.5: The table shows the descriptive results in from the experiment.

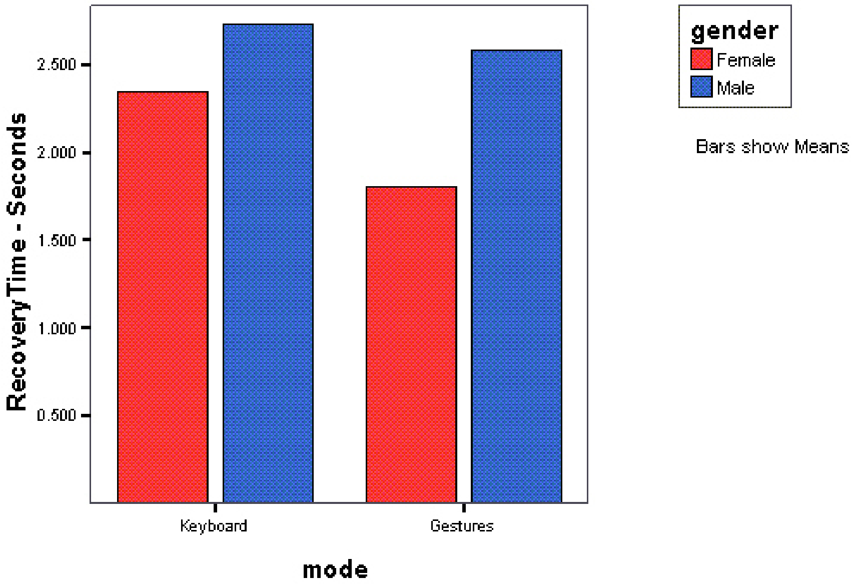

FIGURE 3.14: Graph shows the recovery times for mode, and gender. Overall, recovery time was lower for gestures, as it was for female participants. Times are measured in seconds.

3.3.3.2 Quantitative Results

Significant results were found for the within-participant measures of on *recovery time* ($F_{(1,307)}$ =5.843 at p=.016, power=.674), with mean values for gestures=2.24 seconds, and keyboard=2.56 seconds. Results are shown in Figure 3.14. Significance was also found for *distraction time* based and order of mode presentation ($F_{(1,307)}$ =7.402 at p=.007, power=.774), however, there was an overall lower *distraction time* for gestures than for the keyboard (gestures=4.24 seconds, keyboard=4.4 seconds) but this was not significant between participants ($F_{(1,307)}$ =.072 at p=.778, power=.058). Although this does not provide a high level of predictive power, it does suggest that there was a learning effect present that was reflected in the level of distraction caused to the primary task. A graph for this result is shown in Figure 3.15

There was also a significant gender effect between participants for recovery time, ($F_{(1,307)}$ =19.959 at p=.000, power=.994), with mean values of overall *recovery time* for females=2.07 seconds, and males=2.65 seconds, shown in Figure 3.16.

3.3.3.3 Qualitative Results

Satisfaction. We measured user satisfaction according to results of our post experiment survey, showing that overall, gesture was the preferred mode of interaction for the

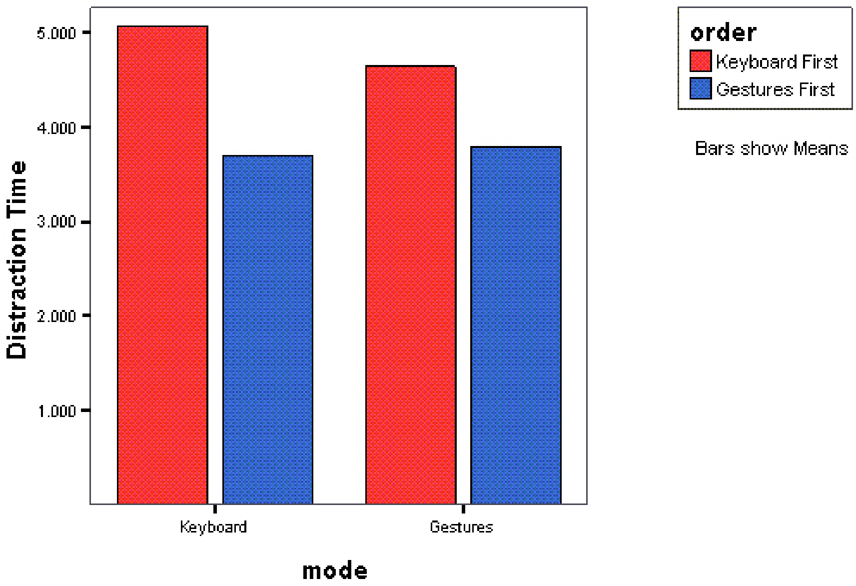

FIGURE 3.15: Graph shows that the distraction times for gestures were overall lower than for keyboard. When keyboard was shown first, participants appeared to adapt quicker, however this would have been due to their familiarity with the keyboard over the gestures. Overall, gestures caused less distraction. Times are measured in seconds.

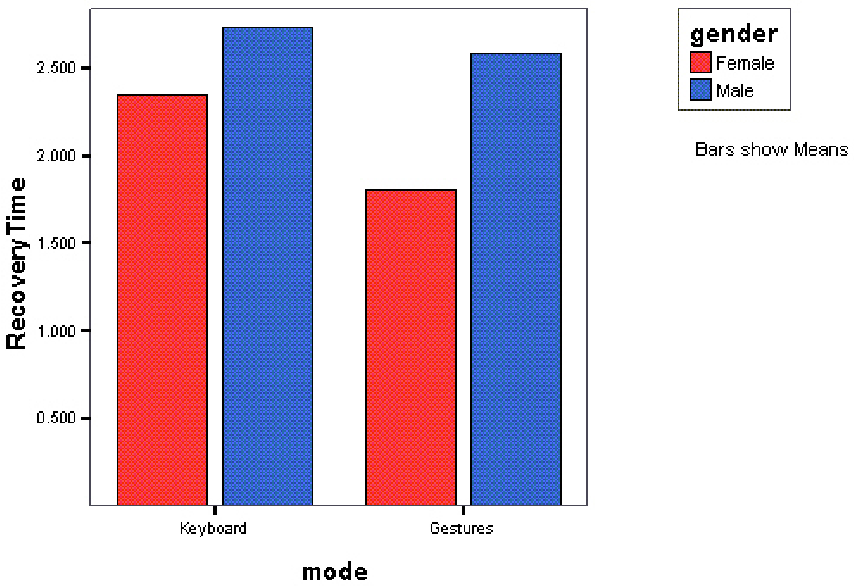

FIGURE 3.16: Graph shows Recovery times for gender and interaction mode, with both genders experiencing faster recovery time with the gestures than with the keyboard interaction. Times are measured in seconds.

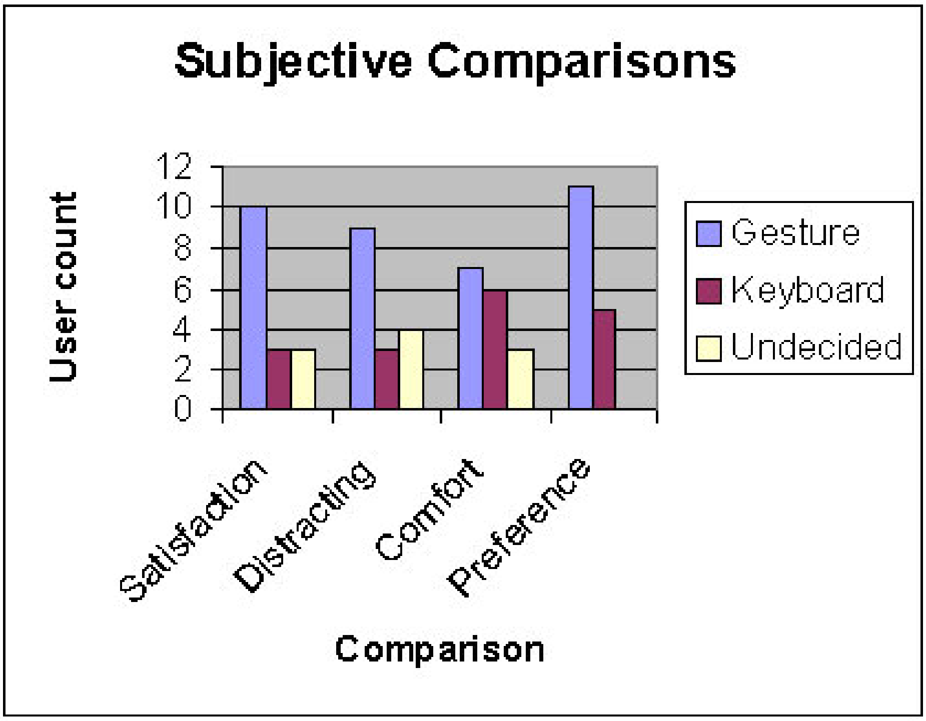

FIGURE 3.17: This graph shows the results of the qualitative data obtained from the post-evaluation questionnaires.

subjects. The results are presented in 3.17. Post experiment interviews revealed that all of the subjects would prefer to gesture when the keyboard was not close at hand. The results on comfort were split, likely due to the extra movements required to gesture, which seemed excessive to some. Gestures can lead to arm tiredness over extended usage, however in a real situation, it would be unlikely that anyone would gesture to the extent required for an experiment. However, the trade off in a real-world situation was said to be worth while in the open ended interviews that followed the debriefing for the experiment.

Interaction Errors There were a total of four errors made with the gestures, and four errors with the keyboard between four participants. 12 participants made no interaction errors.

3.3.4 Discussion

Physical interaction requirements and task completion. Our results showed that, in this set up, gestures took slightly longer to complete than the keyboard interaction. However, in a real scenario, if a device is not within reach, the larger interaction of gestures would lead to quicker interactions, especially if devices have to be located.

Thus, the number of steps to complete that simple, final one gesture press of a button may increase dramatically. Gestures have the potential for detection within a large area, based on the visual range of the camera, thus maintaining a persistent two-step interaction to complete a given secondary task.

Vision requirements and task recovery time. A perhaps more telling result than task completion time is task recovery time. The results show that task recovery time is significantly shorter when using gestures than when using the keyboard. Observations of participants during the experiment suggested that this led users completing secondary tasks without taking their eyes off their primary task and may account for the significant difference in task recovery times when using gestures: less attention is required to attend to the control of the secondary task; more visual focus remains on the primary task. Hence, the threshold for recovering a visually-oriented primary task focus is reduced when using gestures.

Gestures: Ease of use. Since the number of errors that were observed for the gestures were equal to the errors with the keyboard interaction, we concluded that despite the novelty of gestures as an interaction method, they were as easy and accurate to use as the more familiar keyboard interaction.

Gender and Recovery Results show that female participants were more quickly able to recover from the distraction of the secondary task than male participants. This supports existing theories that females are better able to deal with multitasking situations than males.

3.3.4.1 Note on Statistical Analysis

Thought there a variety of different models that could have been used to conduct this analysis, the one conducted in this chapter supports our claims that gestures are less disruptive in multitasking situations. Though the power ratings for our results were not high, the trends and observations do support this result. However, we note that conducting laboratory studies on users does not represent the ideal scenario for determining real usability results for any interaction technique. A more telling approach would be to provide the opportunity to use gestures in a real-world scenario, for a variety of users, over an extended period of time. Results from these empirically based lab studies do however provide evidence to suggest that gestures are a less distracting technique, however real-world, ethnographic or qualitative approaches would provide a much more complete picture that would enable us to make more decisive conclusions about the effectiveness of different interaction techniques. This is an approach that will be explored in future studies.

3.4 Summary

While Wexelblat and others have criticised the use of semaphoric gestures as an interaction technique, results of an experiment suggest that there are significant benefits to semaphroic gestures for secondary task interaction. We also argue that there are benefits in terms the consistent number of steps required to gesture at a camera, whereas the use of direct input device controllers can introduce increased delays in completing the secondary task and potential degrade recovery of primary task focus. Task recovery time is of critical interest in notification systems research conducted by Czerwinski et al. (2004) and our results suggest that significant improvements are possible when using gesture over function keys for reducing task recovery time. Our results also suggest that interaction mode is a significant factor for assessing interaction performance with ambient or secondary task systems. In the next chapter, we further explore the domain of gestures in an empirical study designed to provide us with a more detailed understanding of the different characteristics of multitasking while investigating user tolerance for errors in gesture recognition.

Chapter 4

Investigating User Tolerance for Gesture Recognition and Performance Issues

> *"Each problem that I solved became a rule, which served afterwards to solve other problems."*

Rene Descartes

4.1 Introduction

In our previous chapter, results from an experiment suggested that there was indeed a functional utility for semaphoric gestures. In that experiment, error rate was held constant, where we did not simulate any recognition errors. We next embarked on a study to determine the level of recognition error that a user can tolerate. Since current state of the art vision technology cannot yet achieve 100% accurate recognition, we set out to determine what level of recognition error users would tolerate. To design an appropriate scenario for this experiment, we conducted a participant observation study to explore tasks, and interaction scenarios that could inform our scenario design. That study is presented in detail in Appendix C, and the applicable results which informed the scenario we used in the current study are discussed in Section 4.3 of this chapter. As in the previous experiment, presented in Chapter 3, the study described in the current chapter considers gestures from the interaction perspective, and supports the human perspective of interaction research. However, in this work, we began to apply our knowledge gained from previous experiments to guide this research.

For the error tolerance experiment discussed in this chapter, we extend the work presented by Beaudouin-Lafon (2004) and propose an interaction model to create a framework for guiding our evaluation of gestures. The model represents lessons learnt and experience gained from our previous work with gestures in Chapter 3. We could now understand gestures in terms of their interaction context, system performance measures and the goals of the users. We apply our experience and define interaction context in terms of the physical layout of the interaction space, system performance in terms of the accuracy rate of the gesture recognition system and user goals in terms of task characteristics. With this structure in place, we began our investigation to determine what level of accuracy is required for a gesture detection system to be both tolerated and experienced as useful, and in what contexts might gestures be more appropriate over alternative, physical input mechanisms?

We used the Wizard of Oz (WoZ) methodology for this experiment, where we explore user tolerance for errors in gesture recognition systems, described by the interaction model. We also demonstrated how researchers and designers can apply these results to assist in determining if gestures will enhance an interaction scenario. We continue our investigate of semaphoric gestures in this work, where hands are used to sign or signal commands to the computer and discuss how our proposed interaction model can be extended to inform future evaluations. In the next section, we present related work that explores user tolerance for computer interactions, followed by the details, results, and conclusions of our experiment. A short version of this study appeared in the Conference on Advanced Visual Interfaces, 2006 Karam & schraefel (2006).

4.2 Interaction Model

We discuss the three elements proposed for our interaction model for investigating gestures, and their role in influencing user tolerance for recognition errors. We refer to Beaudouin-Lafon (2004) definition of an interaction model for this research, as having the following function:

> The purpose of an interaction model is to provide a framework for guiding designers, developers and even users (in the context of participatory design) to create interactive systems. An interaction model is thus more operational than an interaction paradigm and can be used directly by designers.

We present three main elements of our proposed interaction model, and their role in providing a framework from which we can design our interactions to determine user tolerance for gesture system recognition errors.

Interaction context. This experiment presented in this chapter investigates user tolerance and satisfaction with gesture interactions by varying physical aspects of the interaction context to create two scenarios: ubiquitous and desktop computing. Our scenario uses a camera as the input device for the gestures, while providing a direct-input device (a keyboard) at a distance from the user as a back-up or alternative device for controlling the secondary task. We measure user's tolerance as the number of times they choose to use the keyboard instead of the gestures. In the desktop scenario, the keyboard is located in front of the participant and the monitor used in the experiment (see Figure 4.7). In the ubiquitous scenario, we physically extend the desktop metaphor so that the keyboard is located away from the participant, thus simulating the distance style interaction of a ubiquitous computing scenario.

System performance. To investigate system performance and the effects on usability, we considered the sensitivity level of the recognition system —highly sensitive leads to false positive errors, while low sensitivity leads to false negative errors —as well as the time taken to process and respond to a gesture command. In this experiment, we concentrate on error rates as the key variable for measuring user tolerance.

Users' Goals For this experiment, we investigated a multitasking situation, where the participants work on non-computer primary tasks, while controlling computer-based secondary tasks. While there are many different task characteristics that can define or describe user goals, we refer to those determined in our ethnographic study (see section 4.3 later in this chapter, including the cognitive and physical nature or complexity of the tasks, the relationship between the primary and secondary tasks, and the criticality of the secondary task. To create criticality, we imposed a timing constraint for the critical tasks, so that the users' goal was to complete the primary tasks as quickly as possible while controlling the display (secondary tasks). In the non-critical condition, users were to complete the primary tasks without timing constraints while controlling the visual display. We also considered related tasks, where the primary task was dependent the secondary task, and a single-decision task, consisting of a single unit or gesture interaction to complete.

4.3 Exploring Multitasking Characteristics - A Participant Observation Study

In our previous research, we investigated gestures for a variety of interactions, with a primary focus on controlling audio output through music players and applications, discussed in Chapter 3. However, to extend our knowledge about gestures, we wanted to explore different interaction scenarios and tasks for controlling visual displays. To assist with our experiment design, we conducted this ethnographic study, using a participant observation methodology to determine what scenarios we could use to create a more natural environment within our lab setting. We present the details of the ethnographic study in Appendix C, and provide a summary of our results and their influence on our experiment design next.

4.3.1 Multitasking and Task Characteristics

While we gathered a large amount of data from this ethnographic study, its primary function was to assist in determining how to design the tasks for our experimental multitasking situation. We uncovered several task characteristics that support previous research Wild et al. (2004), however our interest for this study is on designing tasks for visual displays. Our study revealed that during leisure times in a home environment, participants would often spend hours seated at a table, where they would perform activities such as reading, or working on hobbies or puzzles. We only considered tasks that could be constrained within single location in the lab to support empirical evaluation. We chose a series of puzzles, games and reading material to simulate leisurely activities as the primary task, and because they require enough concentration to engage the participants and enable a better opportunity to gauge tolerance levels. The only task characteristics we tested in this experiment was criticality, which was created by using a timer to encourage the participants to work against the clock to complete their primary and secondary tasks. We next present the details and results of our current experiment to investigate user tolerance for gesture system recognition errors.

4.4 Experiment

We conducted a Wizard of Oz (WoZ) experiment to explore quantitative, quantitative and subjective aspects of user tolerance, behaviour and perceptions of semaphoric gestures for secondary tasks in a multitasking situation. We chose three variables to test user tolerance and satisfaction in relation to our proposed interaction model.

4.4.1 Experimental Design

The experiment was a 2x2x5 (2 levels of context, 3 levels of task characteristics and 5 levels of error rates) randomised, factorial design with repeated measures. Four different primary tasks were used in each experiment session, described in Table 4.1. Experiment sessions consisted of 4 blocks of primary tasks. Each block had a series of 30 trials (secondary tasks) and participants could choose to gesture or use the keyboard for each. There were a total of 120 trials per session, and each trial could be completed using a gesture or a key press. Sessions lasted approximately one hour. For this experiment, we wanted to explore different types of secondary task interactions to expand on our previous understanding of this scenario 3. We approached this task through an ethnographic study to learn about multitasking situations and gesture interactions with visual displays. Like our previous study, we focus our secondary tasks on ambient displays.

Participants. 46 Students from a second year HCI course were recruited from various disciplines including physics, computer science, biology, chemistry and social sciences and were encouraged to participate in the experiment as part of their course work although this was strictly voluntary. Since gestures are a novel interaction technique, having participants with computing experience assisted in reducing the learning that may be required for less experienced users.

Approach. A pilot study was conducted in advance of the experiment to help us determine which of the variables from the interaction model would be most effective in determining user tolerance. While the pilot study considered many additional variables that influenced tolerance, results suggested that we control three independent variables:

1. Interaction scenario. An alternative input device is located close to the user (desktop condition) or at a distance from the user (ubiquitous condition) to give the participants the chance to express their choice to use an alternative input mode when tolerance levels for gestures are exceeded. Our hypothesis stated that users would be more tolerant of errors in the ubiquitous scenario than in the desktop scenario.

2. System performance and recognition error rates: The average number of errors that the system made during a single block of trials. We hypothesised that user tolerance for recognition errors would be greater in a ubiquitous computing scenario than in the desktop scenario, and should exceed 10% minimum rates for both scenarios.

3. User goals and task criticality: We imposed an element of criticality on the primary tasks by displaying a timer for these non-computer tasks and instructing the participants to complete them as quickly as possible. Our hypothesis stated that the more critical the task, the less tolerant users would be with recognition errors.

4.4.2 Method

The experiment was conducted at a desk with a monitor placed in front of the participants. Primary tasks were located on the desk within reach of participants. A keyboard was positioned either at the far right side of the desk or in front of the participant.

	Primary Task	Secondary Task
Related Tasks	1. Jigsaw Puzzles	Advance slides to get instructions on which puzzle to do and to indicate when you begin and again when you are done
	2. Reading: Sections of a book are assigned for each trial.	Advance slides to get instructions on which section to read and to indicate when you begin and again when you are done
	3. Stix and Balls: Build structures using the magnetic balls to connect the stix.	Advance slides to get instructions on which shape to build and to indicate when you begin and again when you are done
Unrelated Tasks	4. Assorted crossword and brain-teaser puzzles	Gesture to start or stop background music when an alert is issued - Notification system interaction

TABLE 4.1: Table describes the different tasks used in the experiment, ordered as they were presented to the participants.

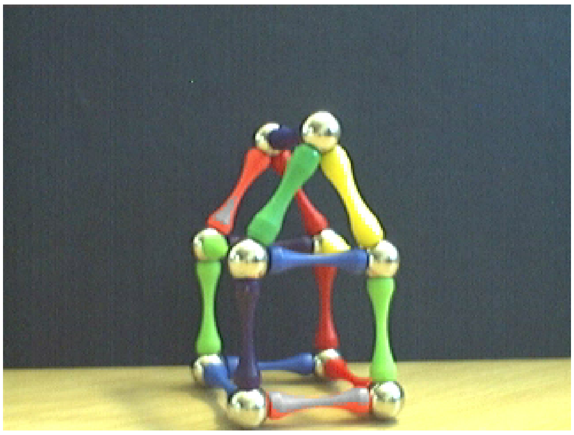

FIGURE 4.1: The picture shows the structure that participants were to build for one of the primary tasks.

We refer to these scenarios in terms of the application domains they could represent based on our classification of gesture interactions, with distance interaction representing a ubiquitous style of computing, and the keyboard located in front of the monitor, as in a desktop computing scenario. Figure 4.7 shows both conditions; on the left, the desktop condition with the keyboard located in front of the participant and on the left, the ubiquitous condition with the keyboard away from the user at the far right side of the desk. Participants could choose to use either the gesture or the keyboard to control the secondary tasks.

Primary and secondary tasks. The primary tasks were non-computer activities and included jigsaw puzzles, reading, building geometric structures (see Figure 4.1) and assorted puzzles (crosswords and brain teasers). Table 4.1 describes the 4 tasks, and their characteristics identified for the experiment. Several tasks were performed from a single activity for each block of trials: there were 6 different jigsaw puzzles, 6 building tasks, 100 reading tasks, and 15 brain teasers to work on. Instructions were presented as a slide show on the visual display and were advanced by the participants during the

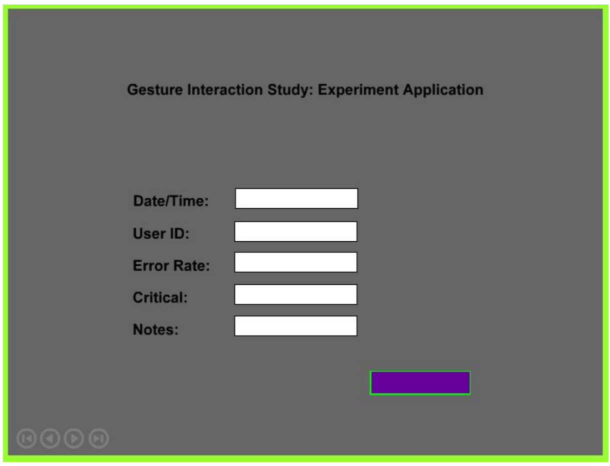

FIGURE 4.2: Screen shot of the instruction page for the experiment.

FIGURE 4.3: Screen shot of the instruction page for one block of tasks, in this case, it describes the jigsaw puzzle tasks.

experiment as the secondary task. A series of screen shots are presented in Figures 4.2, to Figure 4.6. The secondary tasks involved advancing slides on the display. The slides presented instructions about each primary task activity they were to perform (see Figure 4.5). The participants could advanced the slides using either a single gesture or by pressing the right arrow key on the keyboard.

Gestures A single gesture was used for the experiment, consisting of a horizontal hand movement as in a wave performed in front of the camera. We chose to use a single gesture for this experiment to reduce confounds due to having to memorise or learn multiple gestures.

Apparatus and set up The apparatus consists of a PC running Windows XP, and a flash program to run the slides used for the experiment. Slides were displayed on

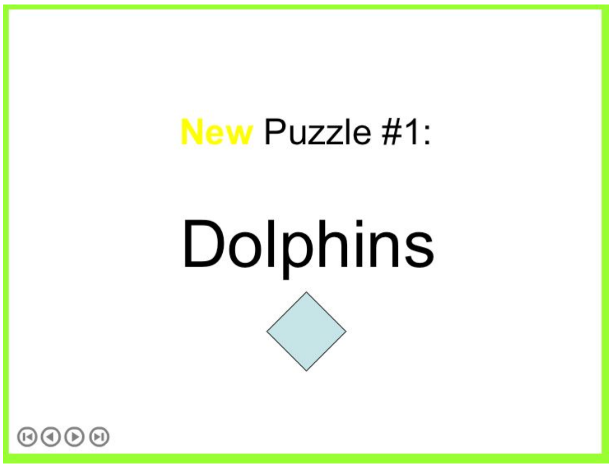

FIGURE 4.4: Screen shot of the instruction page for the individual puzzle which participants were to complete.

FIGURE 4.5: This screen was displayed after participants advanced the slides to signal the start of a task.

FIGURE 4.6: This screen was displayed after participants signaled the end of a task.

FIGURE 4.7: In this picture, the left image presents the desktop condition, and the left image presents the ubiquitous condition.

a flat panel monitor located in front of the desk, and we used a keyboard to provide an alternative interaction mode to gestures and a web camera to monitor the gestures. A second camera recorded the sessions and an iMac G4 computer running OSx was used by the Wizard to control the gestures and issue the feedback indicating successful recognition.

Procedure. Participants filled in a consent form and a pre-evaluation questionnaire. The researcher then explained the tasks, described in Table 4.1, and demonstrated the use of the gesture system. Participants were instructed to gesture using either their right or left hand. A coloured paper wristband was placed on either the right or left hand —as chosen by each participant —to enable the Wizard to track gestures using the camera. Each participant was given as much time as they needed to practice using the gestures. The participants were told to use either the gesture or the keyboard to advance the slides (secondary tasks) and complete the trials. The experiment concluded with a post-evaluation questionnaire and an informal interview.

4.4.3 Variables

Independent Variables

- Between-Groups - Error Rates: 0%, 10%, 20%, 30%, 40%. Errors were distributed at pre-determined intervals throughout each block and the error rates were based on the number of errors present for each set of tasks within a block, and held constant for each session.
- Interaction Context: The keyboard was placed in one of two locations to create the two interaction conditions: In front of the monitor for the desktop condition and at the far right side of the desk for the ubiquitous condition.

- Task criticality: The critical task condition presented a timer on the screen and required the participants to complete the tasks as quickly as possible, while the non-critical condition did not.
- Within-Groups - Tasks: We use 4 separate blocks of trials, each consisted of several tasks to work on for the duration of the session (see Table4.1).

Dependent variable Tolerance: We calculate user tolerance as the percentage of gestures used to advance the slides compared to keys pressed during the trials, such that the greater the number of times a participant chose to use the keyboard represents a lower tolerance level for errors in gesture recognition. Tolerance was measured for each of the four tasks used in the blocks.

Subjective and qualitative data. We recorded participant responses from the post-experiment questionnaires to obtain the following subjective results:

- User satisfaction or frustration with the gestures
- User confidence in the gestures: Rated from low to high
- Overall impression of the system: Rated from terrible to wonderful
- Confidence in the gesture system: Rated from low to high
- Perceived accuracy of the system: Rated from low to high.

We observations participants during the trials and conducted interviews after the sessions to obtain qualitative data. We discuss the subjective ratings in the discussion section, along with observations from the pilot study, and results from the experiment sessions.

4.4.4 Results

We planned to use the complete set of error rates (0-40%) in both the desktop and ubiquitous computing conditions, however we discovered that once users began to experience any error rate in the desktop computer conditions, tolerance levels were so low, that most preferred to use the keyboard over the gestures. Thus, we did not complete the trials in the desktop condition for error rates over 10%. We ran a repeated-measures analysis of variance (ANOVA) on all of the data for an incomplete factorial analysis. Results are discussed next.

4.4.4.1 Within-Group effects

We found several significant results for the within-group tests and discuss each in turn.

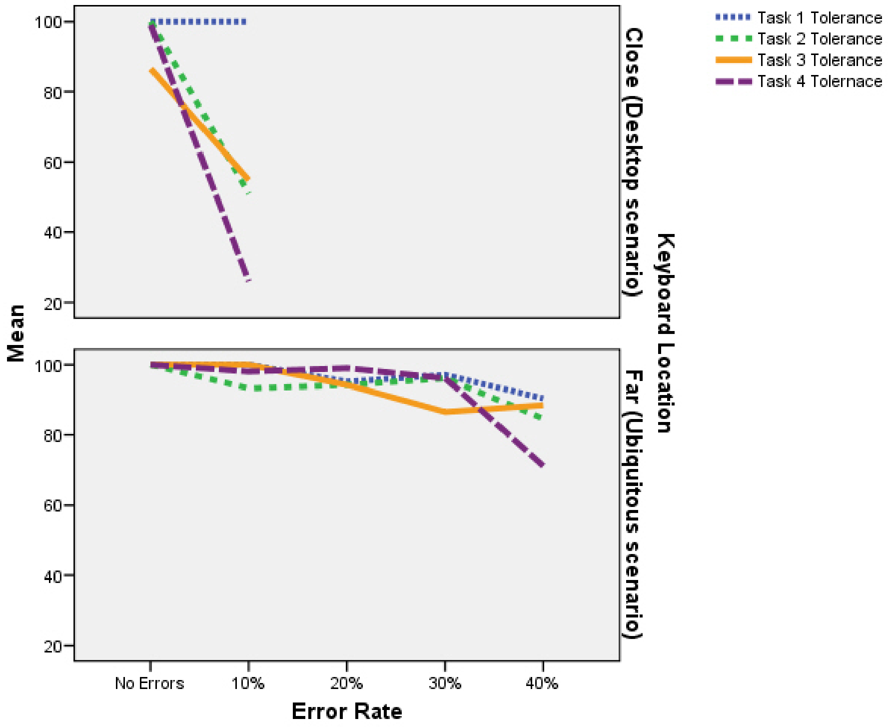

FIGURE 4.8: The profile plot shows the tolerance levels for the different tasks (within-group) in the two keyboard locations.

Tolerance. Tolerance levels between the four tasks were found to be significant ($F_{(1,14)}$=8.995, p<.01), showing a trend for decreasing levels of tolerance with each task in succession (means: task1 97.53, task2 88.46, task3 87.23, task4 84.20).

Tolerance levels for the four tasks showed a within-group interaction effect for task characteristics, with an increase in tolerance in the fourth task in the critical condition, but not in the non-critical condition ($F_{(1,14)}$=5.024, p<.05). Tolerance also shows an interaction effect with the interaction contexts, with a slight increase in tolerance level in task 3, but then decreasing for task 4 in the ubiquitous condition ($F_{(1,14)}$=6.011, p<.05), shown in Figure 4.8. Tolerance levels also appear to interact with error rates and interaction context ($F_{(1,14)}$=2.916, p<.05), showing that in the ubiquitous condition, the increased tolerance level in task 3 occurs in the 10% error condition.

4.4.4.2 Between-Group Effects

We found several interesting results in for the between-group analysis shown in Table4.2. First, significant results were shown for error rates, keyboard location and timing. To investigate the interaction effects, we conducted a second ANOVA to explore the two interaction contexts.

Tests of Between-Subjects Effects

Measure: tasks
Transformed Variable: Average

Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Intercept	701256.245	1	701256.245	1655.035	.000
ErrorRate	9128.837	4	2282.209	5.386	.008
Timed	6053.933	1	6053.933	14.288	.002
KBD	7572.347	1	7572.347	17.871	.001
ErrorRate * Timed	4007.535	4	1001.884	2.365	.103
ErrorRate * KBD	5270.202	1	5270.202	12.438	.003
Timed * KBD	3273.160	1	3273.160	7.725	.015
ErrorRate * Timed * KBD	2746.163	1	2746.163	6.481	.023
Error	5931.953	14	423.711		

a. Computed using alpha = .05

Table 4.2: The table presents the results of the between-group ANOVA.

Tolerance. Results from this ANOVA show that in the ubiquitous condition, there were no significant differences in the tolerance levels for any of the independent variables of error rate or task characteristics, suggesting that users are extremely tolerant of recognition errors in the ubiquitous condition. However, this does not occur in the desktop condition.

The desktop condition reveals significant differences in tolerance for error rates $F_{(1,4)}$=30.993, p<.05) such that tolerance significantly decreased by the 10% error condition. Timing is also significant ($F_{(1,4)}$=19.835, p<.05), showing a lower tolerance in the timed condition (mean=89.43) than in the non-critical condition (mean=97.03). We also found an interaction effect for error rate and timing ($F_{(1,4)}$=16.857, p<.05,) where tolerance appears to converge at the 0 error rate, but diverges in the 10% error rate condition, with lower tolerance levels in the critical condition.

4.4.4.3 Subjective Results

Results suggest that there is a significant correlation between users overall tolerance level and their confidence in the gestures (.578 at .01) and overall impression of the system (.503 at .05). Error rates showed a negative correlation to user confidence (-405 at .05) and a positive one to perceived accuracy (.617 at .01) however, no correlation was noted between error rates and satisfaction, supporting our hypothesis that error tolerance is more dependent on other factors than on error rates alone. Mean values for satisfaction, overall impression and confidence are presented in Table4.3.

Report

Error Rate		Satisfaction	Estimated Accuracy	Overall Impression	Confidence	Overall Tolerance
No Errors	Mean	6.20	8.10	6.67	8.00	93.0215
	Std. Deviation	1.687	1.853	1.118	1.054	12.67485
10%	Mean	6.64	7.07	6.27	6.36	85.0799
	Std. Deviation	2.061	1.817	1.348	2.170	17.66188
20%	Mean	5.10	6.30	6.20	6.20	92.7598
	Std. Deviation	1.729	1.889	2.280	2.616	9.15338
30%	Mean	5.50	6.00	5.80	6.17	94.5837
	Std. Deviation	2.588	1.549	1.789	1.472	7.19492
40%	Mean	7.17	5.67	5.83	5.67	90.1433
	Std. Deviation	1.941	1.966	.983	2.251	8.25462
Total	Mean	6.13	6.80	6.22	6.57	90.3760
	Std. Deviation	2.018	1.939	1.416	2.094	12.89336

TABLE 4.3: The table shows the mean values of the subjective results for all participants.

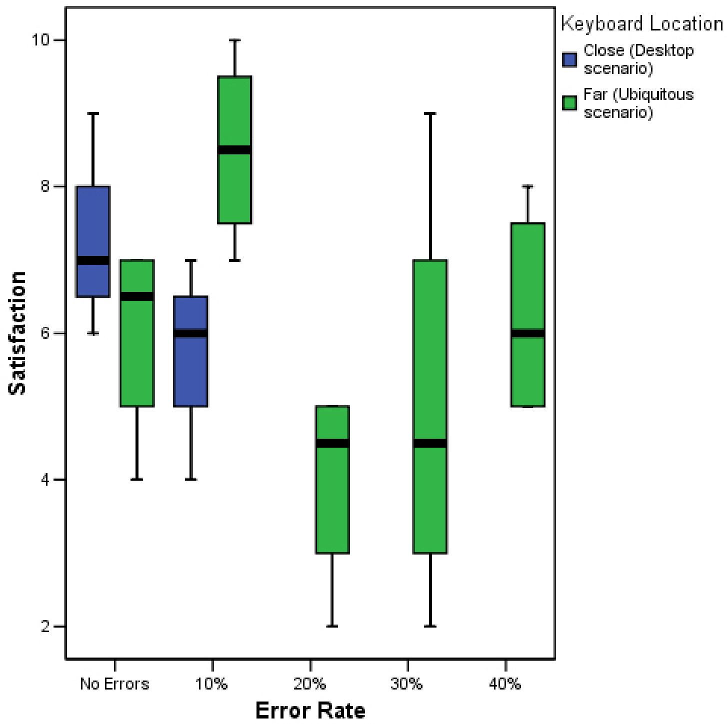

FIGURE 4.9: The graph shows error bars for user satisfaction ratings according to error rates and interaction scenario.

Keyboard Location		Satisfaction	Estimated Accuracy	Overall Impression	Confidence
Close (Desktop scenario)	Mean	6.75	7.67	6.00	6.42
	Std. Deviation	1.288	2.015	1.333	2.151
Far (Ubiquitous scenario)	Mean	6.50	7.60	7.00	7.80
	Std. Deviation	2.369	1.578	1.069	1.619
Total	Mean	6.64	7.64	6.44	7.05
	Std. Deviation	1.814	1.787	1.294	2.011

TABLE 4.4: The table shows mean subjective results for the participants in the 0 and 10% error conditions.

Satisfaction. Participants rated their satisfaction with gestures on average higher in the ubiquitous condition than in the desktop condition, however, there was little difference noted between error rates. The critical condition did show a slightly lower rating for satisfaction than for the non-critical condition (mean timed: 6.07 mean not-timed: 6.14) as shown in Figure4.9. Results suggest that a more satisfying interaction experience can be achieved provided that the time and effort taken to perform the gestures does not outweigh the benefits.

Impressions Overall impressions were rated slightly higher in the desktop scenario than in the ubiquitous condition, however this may be due to the additional error rates seen by these participants. When we look at the ratings and compare only the 0 and 10% error rates we do see that mean overall impression for the ubiquitous scenario (mean=7.0) is higher than for the desktop scenario (mean=6.0) as shown in Table4.4.

4.4.5 Discussion

Interaction context. We considered two interaction contexts based on the location of an alternative input mode; desktop computing when a direct controller is directly in front of the interaction space, and ubiquitous computing when the keyboard is located away from the primary task. In this experiment, participants seemed to instinctively chose to use the keyboard over the gestures when it was close at hand. In this case, gestures not only provided less control than a direct input device, but took more time to execute than a key press. We extend these results and suggest that the gestures are most appropriate for situations in which the user sees a distinct benefit in having access to distance interactions or for extending the flexibility of desktop interactions.

Error rates. While 100% recognition accuracy of computer vision technology is not yet possible in everyday computing technology, this study suggests that error rates can potentially reach 40% before user tolerance levels fall off in the ubiquitous computing scenario, however in the desktop scenario, tolerance drops significantly when users have access to a more familiar direct input devices such as the keyboard. While error rates are a significant factor in influencing tolerance, the results suggest that the interaction scenario is more influential in determining user satisfaction and tolerance levels for error. This would imply that despite the imperfect recognition capabilities of perceptual computing input technology, gestures could still provide benefits and satisfying secondary task interactions for ubiquitous computing scenarios.

Task characteristics. While we consider several tasks in this experiment, the main characteristic investigated is the level of criticality of a task. While we found only a slight increase in the tolerance for errors in the non-critical tasks, results suggest that for tasks that require a greater degree of precision or accuracy, gestures may not be an appropriate control for several reasons. First, the lack of precision in gesture recognition may not provide appropriate support for tasks that require a high level of accuracy and second, due to the additional delays of processing perceptual input, a task could take longer to perform than when using a direct input device. Finally, although we did not specifically investigate the different task characteristics in this experiment, our results suggest that there may be differences due to task characteristics in the tolerance users have for errors in gesture recognition, however while the results also suggest that these may effect tolerance levels for errors, further investigation would be required to understand these differences.

4.5 Qualitative Analysis

This section discusses the qualitative results obtained through observations during the pilot study, the experiment, and the post-experiment interviews with the participants. These are organised according to the interaction model.

Interaction context. In this study, we considered two interaction contexts, desktop and ubiquitous computing. We note that in the pilot study, we tested error rates up to 60% in the ubiquitous condition before participants exhibited frustration with the gestures and chose the keyboard. In addition, we found that the gesture recognition responses could take over 4 seconds before participants would be forced to use the keyboard, and many of them continued to user the gestures in spite of this long delay. Again, this suggests that interaction context is a significant factor to consider when trying to determine if gestures are an appropriate interaction technique.

Alternative interaction modes. When considering gestures as an interaction technique for any context, most devices come equipped with remote controllers, buttons or other direct-input controls so that in case of failure, there is an override system. This was reflected in our use of the keyboard as an alternative input mode. Thus, we note that providing users with an override to the gestures should be considered as an essential feature of gesture interactions.

4.5.1 System Performance

Gestures and error type. We intended to investigate both false positive (FP) and false negative (FN) errors for the experiment. However, during our pilot studies, we note that in the FP errors caused participants to drastically alter their movements during the session so that their gesturing hand was kept out of view of the camera. This involved using only one hand to complete the primary tasks, or careful attempts at moving the gesturing hand to avoid the camera. Two participants asked if we could turn off the gesture system when they became frustrated with the errors. The choice to exclude FP errors from the experiment however was made since most gesture recognition systems should be able to adjust the sensitivity of the recognition to best suite the type of interaction, and because using both FP and FN errors would limit the amount of control we would have had over the experiment for our analysis.

System response Based on our observations and interviews, we noted that in the ubiquitous scenario, system response speed seemed less important than in the desktop scenario. This may be due to the perceived cost of having to move away from the primary task to use the keyboard compared to the time required to wait for the system to respond or recognise a gesture. While we did not test response speed for the actual experiment, we do note that this does play a role in conducting a trade-off analysis when designing gesture interactions. Even if the user feels they can perform the task using an alternative input device in faster time that it would take to perform the gesture, other factors can still influence their decision to use an alternative device. We note that additional factors can influence this choice, and include the goals of the user, the level of concentration they want to maintain on their primary task and the extent to which they want to avoid moving or reaching to access the controls of a secondary task.

4.5.2 User Goals

Eye contact and the camera. During the sessions, participants made eye contact with the camera, mainly in the beginning of the session when learning to use the system. As confidence grew, eye contact with the camera decreased and participants would maintain visual focus on the primary task during a gesture. During an error, eye contact with the camera resumes until the gesture is recognised. This appears to be a natural behaviour during the familiarisation period, and when the system is not responding as expected. This observation suggests that some form of feedback, which we refer to as reflexive feedback —where the computer provides a visual representation of what it is processing —could potentially benefit users during error or learning periods with the system. We have completed a study to investigate reflexive feedback and how it can enhance the interaction and discuss this in Chapter 6.

Task characteristics. While we do investigate task criticality as a factor in the experiment, our attempt to understand several other characteristics of tasks was only briefly addressed. The timing used in the experiment only slightly increased the critical nature of the tasks, however if we had used a more serious constraint, where there would be something at risk for the user, then we note that gestures would not have been an appropriate interaction technique. For example, if a task required any precision or presented any serious risks, then common sense would suggest that the imprecise nature of gesture interactions would not be an appropriate choice for interaction technique in any scenario. However, there are always additional issues to consider that are specific to each interaction context. For example, in a situation where there were no alternative modes of interacting with a display, such as public kiosks, where there may not be space for input devices such as a mouse or keyboard, or where screens are not augmented with touch interfaces, even gesture interactions with a high error rate could provide some level of control for the user.

Handedness Most of the participants were right handed and most chose to use their dominant hand for the gestures however out of the 46 participants, only 5 used their non-dominant hand for the interaction. These participants stated that this was more convenient during tasks such as writing, which required them to use their dominant hand. In considering handedness, performance may increase when using the non-dominant hand for gesturing, enabling the participant to maintain more focus on the primary task.

4.6 Applications for HCI

These results can be applied to several areas within HCI. As an extension to our previous study that considered gestures for secondary task interactions 3, our current results provide additional knowledge about the relationship between the interaction contexts and the appropriate usage scenarios of gestures. First, while gestures can potentially reduce the effects of interruptions during multitasking situations, interaction contexts also play a role in affecting user tolerance, performance and satisfaction with the interaction technique. So while a gesture system with 100% recognition accuracy may not be appropriate for desktop interactions, systems with error rates as high as 40% accuracy may still provide a benefit to users when they require non-critical, distance interactions. Also, even though our critical task scenario was not truly critical, results did suggest that the higher the level of criticality, the less appropriate gestures will as an interaction technique. Results from this study also suggest that there is value in using gesture systems for single-step task interactions, however, we could conduct further investigations into understanding user tolerance for errors when the interaction requires multiple steps, or when placed in a real world setting. However, we have addressed three basic elements of an interaction model for gestures and shown how they play a part in influencing user tolerance and satisfaction with the interaction.

4.6.1 Future Work

This research began to address gestures evaluations from our proposed interaction model. This model provides a framework for evaluating gesture interactions, and is extended to address future experiments. Observations from this experiment suggests that a form of reflexive feedback in gesture recognition systems could assist users by providing kinematic information while gesturing. In addition, quantitative results about the potential benefits of using gestures for notification system interactions could lead to a potential contribution to Link-Up, a claims library, discussed by McCrickard & Chewar (2005): Our results potentially address three of five HCI challenges in designing gestures for cognitive system interactions and supporting the movement toward a science of design. These challenges include demonstrating preliminary contributions to requirements engineering methods for designing gesture systems, proposing a set of measures for predictive modelling of gesture interactions, and developing a conceptual framework for design reuse for gestures.

4.7 Summary

In this study, we explored user tolerance for errors in recognition of bare-hand, semaphoric gesture using computer vision. The study investigated gestures from the perspective of our proposed interaction model, which identifies interaction contexts, system performance and user goals for providing a structure to guide interaction experiments. We addressed two main issues: what level of accuracy is required before a gesture recognition system is usable, and under what conditions are these interactions appropriate. Results suggested that users would be satisfied with gestures and be tolerant of error rates of up to 40% before choosing to use an alternative input mode, however this is conditional on several factors. First, when an alternative input device is within convenient reach of a user, users prefer the more familiar direct-input device over gestures. Also, we must consider the trade-off between system performance and interaction benefits when designing gesture-based interactions. Individual task characteristics should be considered as a factor in determining user tolerance and satisfaction levels when deciding to use gestures, as should system performance measures. Since we can determine error rates and other performance factors for recognition systems through user trials or experiments, this information can inform on when gestures would be most appropriate when considered as an input technique. Observations also suggested that some form of reflexive feedback for vision enabled gesture recognition systems could be used as a technique for improving human gesture performance by providing users with kinematic feedback about the state of system during recognition. This reflexive feedback can leverage the natural tendency of users to seek visual confirmation of the status of the recognition system during interactions with gestures. In the next chapter, we present our theoretical framework for understanding and designing gesture interactions.

Chapter 5

A Framework For Researching and Designing Gesture Interactions

"The Past is to be respected and acknowledged, but not to be worshipped. It is our future in which we will find our greatness."

Pierre Trudeau

5.1 Introduction

Over the past two years, we conducted several experiments and studies to explore gesture interactions. The results of these studies provide us with a good understanding of gestures from the perspective of the categories presented in our classification (see Chapter 2). But while our store of knowledge was increasing with each experiment conducted, we decided that the time had come to change our focus from investigating interactions to taking definitive steps towards our goal of obtaining a theoretical understanding of gestures. In this chapter, we presents our theoretical framework for understanding and designing gesture-based interactions and the methods used in its development. This framework serves two purposes: to guide research and design of gestures, and to provide a structure for understanding gesture systems and their interrelated concepts. The framework is structured around the four categories identified in our classification scheme, and the identification of the concepts and relationships that exist between them. It incorporates the qualitative and quantitative results of experiments conducted earlier in this thesis (see Chapters 3 and 4) using a grounded theory approach. We discuss the theoretical framework next, its development using grounded theory, provide a detailed explanation our philosophical perspective on conducting qualitative research, and present a description of the framework, its categories, sub-categories, parameters and propositions.

Frameworks in Gesture Research The term framework applies to a broad range of theoretical and practical concepts, but is generally considered as a structure to guide programming or research activities. Several examples of frameworks are discussed in the HCI literature, describing practical approaches: Boussemart et al. (2004) presents a framework for designing 3D gestures, Latoschik (2001) uses a framework to describe gesture recognition process, and Morency et al. (2005) employs a framework for implementing head nodding gestures. Other frameworks are theoretical in nature: Kopp et al. (2004) informs the analysis and creation of iconic gestures for autonomous agents and Tang & Leifer (1988) presents a framework for conducting task related research on gestures. Our framework presents a theoretical approach to understanding gestures as an interaction technique. Unlike some frameworks in the computing literature, our approach attempts to provide a structure that address the fundamental concepts and components of gesture interactions that spans the computing sciences, and articulates them as a set of manipulatable parameters which can be used as metrics for guiding evaluations and designs of gesture systems.

5.1.1 Structure

The basic structure of our framework consists of four main categories, which were originally derived using our gesture classification (see Chapter 2). These categories were adopted to represent what we consider high-level components of any gesture interaction systems. The framework begins with a verbal description of each category —physical devices, actions, or goals —and identifies individual entities that belong to the category using the appropriate nomenclature. These terms can be drawn from the associated research domains, supporting the inclusion of existing frameworks or taxonomies within our framework. For example, the category of enabling technology refers generally to input devices that are used to implement gestures. However, when describing individual instances of these devices, our framework can incorporate the terms and concepts that are presented in existing taxonomies (Card et al., 1990) to enable the description and understanding of devices from within the context of their domain. With this approach, the framework can evolve to support a theoretical understanding of the relationships that exist between the categories, while ensuring that the relevant theories are addressed.

Given the high-level nature of the categories, the framework provides a breakdown of each into subcategories that begin address lower-level concepts. A final level is represented by a set of manipulatable and measurable parameters to describe identifiable characteristics and features of the interaction. These parameters represent and describe low-level interaction characteristics that affect and influence user interactions and are associated with each of main categories of the framework. Propositions or hypotheses can then be made to reflect our understanding of the relationship between parameters and categories, and then evaluated using empirical methods.

5.2 Developing A Theoretical Framework

We conducted several experiments and studies that contributed the qualitative and quantitative results that led to the development of the framework: Appendix B presents several studies to explore iGesture, Chapter 3 investigates the functional utility of gestures, and Chapter 4 explores user tolerance for errors and error types for different task characteristics. Our approach to designing the framework involved using qualitative data to form the concepts that are represented in the subcategories and parameters, and quantitative data to provide the metrics used to describe the parameters. We next describe our approach to developing the framework.

Approach. Grounded theory (GT) is a qualitative research methodology that is viewed as a valuable approach to conducting research in the computing sciences (Adams et al., 2005; Elliott et al., 2002; Sarker et al., 2001). The framework was developed using an evolutionary process of subsequent applications of grounded theory. This enabled our analysis of the qualitative data which led to the categories and parameters presented in the framework. Qualitative data originated from the interviews, observations and questionnaires obtained through our experiments. This data was analysed using GT, and is discussed next.

Grounded Theory and Qualitative Research Our approach applied grounded theory to uncover the initial structure of the framework and its associated categories based on out analysis of the codes used to organise the literature review (see Appendix A). The basic premise of using (GT) is in the systematic generation of theory from data that contains both inductive and deductive thinking, sometimes referred to as abductive reasoning (Wikipedia, 2006a; Eliasmith, 2004). While this methodology is commonly used in the social sciences for qualitative data analysis, it has also seen extensive use in the computing sciences (see Section 5.2 in this chapter for references). While there are various approaches to GT, the general principal involves the generation of a theory, hypothesis or framework based on a process of repeated sampling, analysing and theorising about the data until a consistent perspective is obtained. Further analysis was applied to the qualitative results obtained through our previous experiments, which informed the current structure of the framework, shown in Figure 5.1.

5.2.1 Method

We note that the notion of GT according to its original authors Glaser & Strauss (1967), differs from a newer approach, where Strauss & Corbin (1998) considers validation as a key component of the method. Our approach is based on a combination of both versions, since we incorporate quantitative data as a method of validating the elements according to Strauss, but rely on the data gathering method of Glaser's version. There are three main components of GT, based on Strauss & Corbin (1998): Concepts, categories and propositions. Concepts are the basic elements uncovered in the open coding phase, categories represent collections of concepts, and propositions are relationships or hypothesis that are made about the concepts. These are discussed next, where we outline that basic steps we used for GT.

- Open Coding - Concepts: This is first of GT, where data from field notes or transcripts are organised into concepts that are observed in the data. As more data is coded, concepts are added or merged to reflect the content. This is an iterative process of comparing, and modifying codes until the researcher has examined all the data.

- Selective Coding: After an initial round of open coding, a core concept is selected that represents a key issues that has been uncovered in the open coding. From this point, a selective process of coding around this core can guide further coding of the data. This sometimes happens after all the data is coded, however it can also be used to inform initial coding.

- Theoretical Coding - Categories: A later stage of coding, where concepts are merged together as they are compared against the rest of the data enables the researcher to uncover more theoretical concepts that emerge as a result of reviewing the bulk of the data.

- Memoing - Propositions: This is referred to as the core stage of GT methodology where the researcher writes up the ideas that have emerged through the coding process, building on the relationships and forming hypotheses that can be tested using empirical methods.

- Sorting: This stage requires structuring the data so that it can be related to others in a format that is used to connect the concepts towards a theory. While we are not at the stage of generating a theory, this part of the process represents the structure of the elements in the framework.

- Writing: This is the final stage, leading to the product of applying GT. In this case, our final product is the framework for understanding and designing gesture interactions. We present the write up in section 5.3 of this Chapter, as well as a structural diagram of the framework (see Figure 5.1).

While our goal is not the generation of a formal theory of gesture interactions, GT can led to the framework we present in this research. Memoing and the organisation and analysis of concepts and categories were based on hand written notes that were take for each of the participants in the studies and incorporated into the framework in several iterations. Continual applications of GT to the framework will ensure that as novel results are obtained, that the framework can be augmented and verified to address new research.

Obtaining and analysing qualitative data. In this research, we use both qualitative and quantitative data in a cyclical approach where qualitative research informs the development of hypothesis, which is in turn tested in empirical experiments, which generates additional qualitative data. This process began with our work on the iGesture system and our analysis of the literature. This process is referred to as the qualitative, positivist research method (QPR), and is a valuable approach for conducting research in the information sciences (Straub et al., 2004). Additional references for qualitative research, grounded theory and their combination within the positivist philosophy are provided (Schloss & Smith, 1999; Myers, 1997; Straub et al., 2004), We next present a detailed description of the framework and its elements.

5.3 A Theoretical Framework

In this section, we discuss the details of our framework for guiding research and design activities that include informing design decisions, and understanding how individual components within systems can influence the interaction. Each category in the framework is divided into sub-categories, or parameters, which represent the individual artifacts that can be altered in a design or further investigated in future studies. The framework is structured around the four categories that are used to classify the research (Chapter 2). Each category is described using the sub categories, and the parameters. Parameters describe the categories in terms that can be measured and evaluated in empirical studies. A diagrammatic representation of the framework is presented in Figure 5.1.

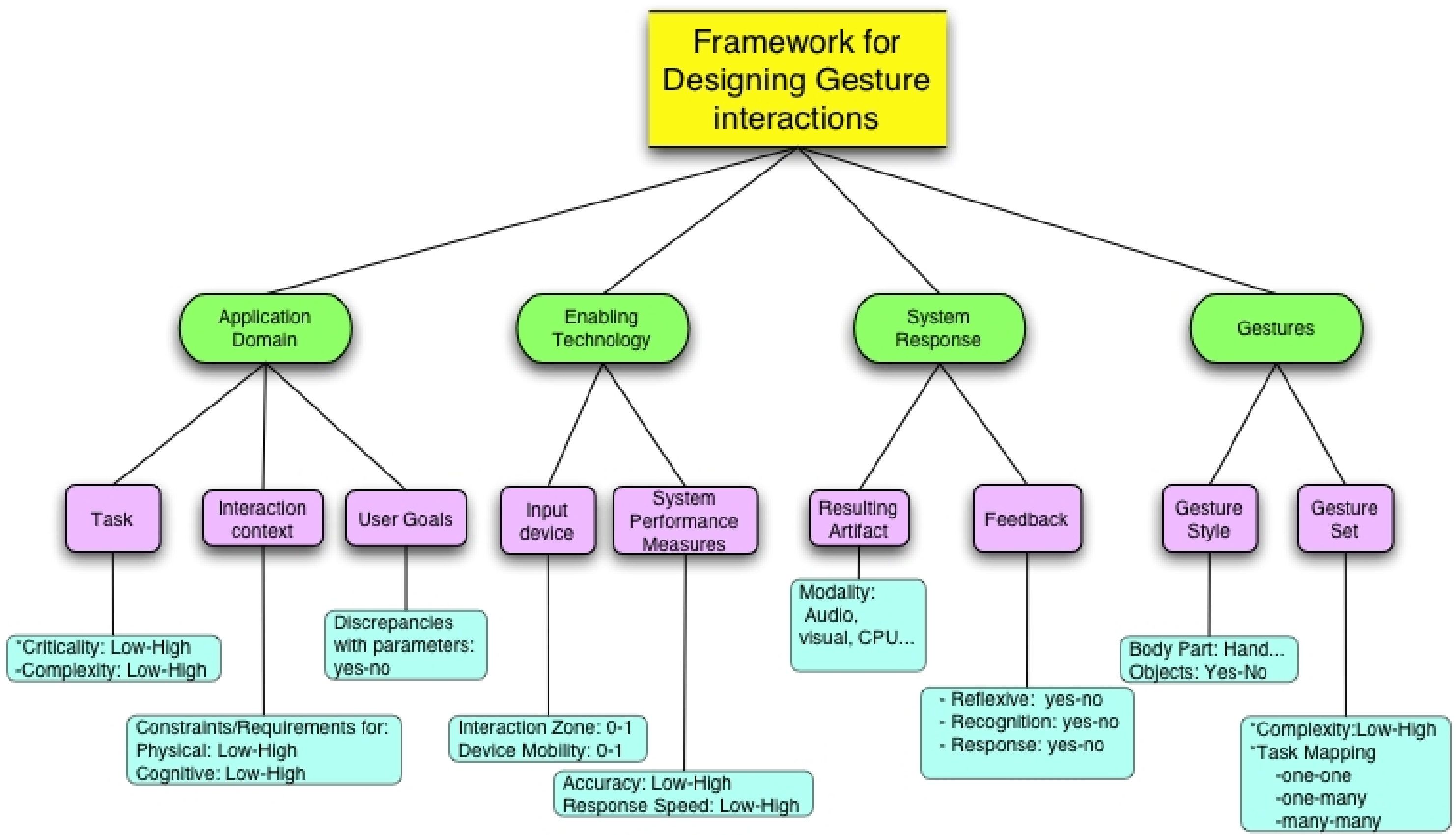

FIGURE 5.1: The diagram represents the structure of the framework. Each element in the top row of the diagram represents one of the four categories in our classification. Sub-categories represent variables within the categories, and parameters propose specific settings that define the categories.

5.3.1 Parameters

Parameters are the lowest level in this framework, and are presented for each of the categories and subcategories within its structure. Parameters are viewed in this model as the manipulatable factors that characterise that category, and for which empirical evaluations can be performed. For example, one parameter under the sub category gesture set is the number of gestures in the set (1,2,...x), while another is the complexity of the gestures (high or low). Each of these parameters have values associated with them to provide designers with specific information about each category. The proposed values or settings that we use to describe each parameter are presented based on the following criteria:

- When referring to a specific device or object, the parameter value is the name or title of the object itself: For enabling technology, parameters include a camera, mouse, DataGlove etc.
- For parameters that are quantifiable, we propose initial ratings of low, medium or high to indicate different values. For example, we refer to accuracy rates or response speeds of a recognition system as being high, medium or low. While this can change with future iterations of the framework, it is an approach taken by Chewar et al. (2004) to enable a general description of describe critical parameter values for notification systems.
- Parameters that are represented as either present or not within a category are valued as 0 or 1: to indicate a setting of on or off, or all or none, present of absent.
- Parameters that represent numeric values are presented using the appropriate value. For example, Gesture sets refer to the number of gestures within a set, and are expressed as that number.

Parameter settings are deliberately general, and represent approximate values that can be determined for any system. While these settings are proposed as preliminary values, they can enable designers or researchers to quickly determine rough estimates of the parameter values for any system with minimum effort. In addition, established ratings within a specific instance of a parameter can also be used to describe performance ratings.

5.3.2 Relationships and Parameters

The framework provides a mechanism for building relationships between the parameters, and applying these relationships to create more effective analysis of systems under consideration. Relationships that are revealed through the framework can support researchers and designers in better understanding the characteristics of the system in terms of the affect they have on the user experience. Though we do not provide a detailed analysis of the relationships revealed in this chapter, we do provide several examples of how the framework is used to organise previous research and the relationships that exist between parameters in Chapter 6. We next present a description of the categories, sub-categories, and their associated parameters and settings.

5.3.3 Application Domains

Application domains tell us a great deal about the context of computer interactions, often implying information about the applicable technologies, tasks, scenarios and contexts that are available within the domain. Within this category, we identify three sub categories that enable us to consider the user and the tasks that are possible within a given application domain: interaction contexts, tasks characteristics, and user goals, which is based on the interaction model presented in Chapter 4. Figure 5.2 presents the key elements of this category, and the parameters used to describe it within the framework.

5.3.3.1 Interaction context

A first element within application domains considers parameters to describe the interaction context. Context, according to Dey & Abowd (1999), refers to:

> any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

We apply this definition and refer to the identified constraints and requirements imposed by the physical or environmental characteristics of the application domain as well as by the specific details of interactions that are conducted within that domain. For example, in a desktop computing domain, the context can be described in terms of the physical state of the user during interactions, such that they will be typically seated in front of a computer for all the interactions within that domain. Context within a mobile domain in terms of the physical state of a user would involve them being free to move around since the device is designed for mobility. We identify two parameters to describe interaction context that are based on physical and cognitive requirements or constraints imposed on the user.

Physical requirements. The level of physical movement required, restrictions on physical movement or other constraints indicate the rating for this parameter. For example, in a desktop computing scenario, we would expect a low amount of physical movement to occur during interactions, while in a ubiquitous domain, movement is not a requirement, however flexibility in the users location is implied, suggesting a medium rating on this parameter, which is based on the low-high scale used in the framework. However in most domains, we must consider the specific interaction scenario before we can determine what physical constraints exist and if they are significant enough to effect the gestures or the other categories in the framework.

Cognitive requirements. The cognitive requirements or constraints are also considered in relationship to their associated application domains. This parameter is expressed as either low, medium or high as determined by the level of cognitive processing required for the interaction. We include perceptual attention (audio, visual) or problem solving resources required for the interaction as the qualitative cognitive requirements, and the amount of cognitive attention required as a quantitative determinant of the cognitive constraints. When more resources are required to complete the task, a higher rating of this parameter is required, as when a greater level of each cognitive process is demanded. Based on the interaction contexts we have investigated for this research, cognitive levels can be determined using the specific requirements of the interaction. For interactions understood within contexts, these ratings can be determined by the researcher or designer based on the perceived level of attention or cognitive processing that is required to complete the interaction.

5.3.3.2 Task Characteristics

Tasks within this framework are considered the primary goal or outcome for which a gesture is intended to control. The task element consists of a description of the task, as determined by designers or researchers. Although we can consider various levels of task descriptions and tasks using a variety of techniques which include hierarchical task analysis or task modelling processes, this element of the framework is intended to enable researchers or designers to focus on any task description that is addressed within the context of the application domain, project or research for which gestures are considered. The decomposition of a task may be addressed using the framework if a high-level description of a task is provided, however this is relevant only to the specific instances or scenarios to which the framework is being applied. Once we have defined the task for which we wish to control using gestures, we can then consider the individual parameters that define the task. We have identified two parameters for characterising tasks that are essential to designing successful gesture interactions: Criticality and complexity. Although these parameters may be related to the interaction context, we consider the explicit values associated with each task, rather than the external factors that are addressed in the interaction context element.

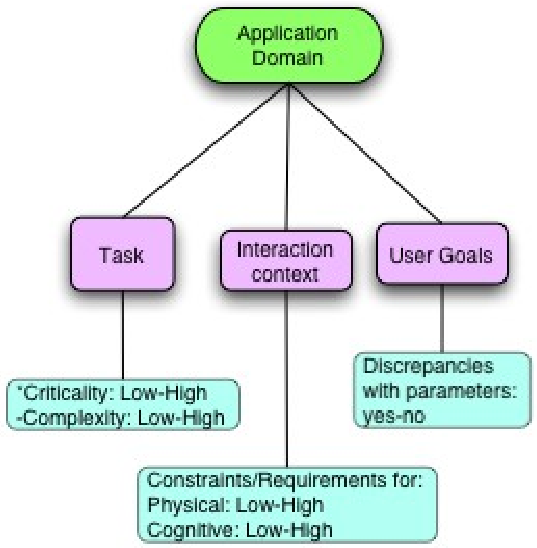

FIGURE 5.2: Task characteristics currently considered in the framework include criticality and complexity.

Criticality. This parameter refers to the explicit level of importance associated with an individual task. A tasks critical level can be expressed as low, medium or high according to the specific tasks rating. This value can be expanded to include additional features that make a task critical: including time constraints, execution constraints, target acquisition constraints or accuracy for example. Additional characteristics can be used to describe the nature of the criticality of a task, however this is dependent on the task itself and should be determined within the context of the interaction scenario. Designers or researchers can further unpack the features of a critical task to enable a finer grain description when one is required.

Complexity. We consider complexity as the number of decisions or steps required to complete a single task rated at low, medium or high. Each rating is based on settings of complexity that can be described as the number of decisions, or the number of steps required to complete a task. For example, single-decision tasks require a single action to complete the task such as selecting the okay button in a pop up alert, and do not require a decision beyond performing the task or no. A multiple-decision task could require the user to select from a number possible options to complete the task such as choosing between the okay and cancel buttons on a pop-up window. Task complexity can also be based on the number of steps and decisions required to complete them. While task complexity can involve in-depth analysis of many characteristics, we approach task complexity from a relative perspective, where tasks are rated based on an overall analysis of the complexity within the context of the system or research under consideration within the framework.

5.3.3.3 User goals

We include user goals as an element that addresses specific requirements of the interaction that are determined by the user. That is, while there are implicit and explicit settings for parameters within this category, the user can choose to override any of the settings previously addressed, or yet to be determined. For example, in a typically non-critical task such as changing the volume of a music player, a user may choose to place a high critical level on this task if they want to ensure that the volume can be changed in a precise, and timely manner with no system errors. In addition, while there may be inherent values set for the physical and cognitive requirements of a given interaction context, the user may wish to change the cognitive requirements from high to low in a case where they have become familiar with the task and no longer high cognitive attention to comprehend the interaction for example. Again, as with the other elements within this framework, we must address specific interaction scenarios in order to set the parameters accurately. This element is based on values of 0 —indicating no conflicts exist between user goals and the other parameters —and 1 to indicate that there is a conflict which can be described by indicating which parameter is to be changed and to what value.

5.3.4 Enabling Technologies

The framework considers the specific characteristics of different technologies or devices used to enable gestures by first identifying the high level description of the input device, and then assessing this in terms of the individual parameters used in the framework. A diagram outlining the elements and parameters within this category is provided in Figure 5.3.

5.3.4.1 Input devices

The first element considered within this category is the actual description of the device used to enable the gesture interactions: where a computer vision enabled system uses a camera as the input device, or a mouse as a direct manipulation input device. This refers to the active device that will be responsible for processing the gesture input. Once we have identified the device, we can next consider the various parameters that enable us to describe the interaction within this framework.

Interaction Zone. Interaction zone refers to the physical space in which users can perform a gesture, based on the capacity of a particular input device to detect gestures. We consider the interaction zone of a direct-input device such as a keyboard or a mouse to be the device itself, where direct physical contact must be made to use it. This

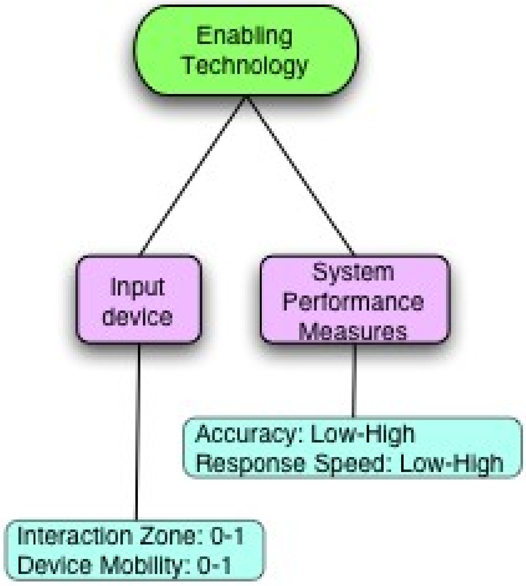

FIGURE 5.3: The structure of the framework surrounding the characteristics of the enabling technologies used in the interaction.

parameter is rated as 0 —indicating that the user must make physical contact with the device —or 1, indicating that there is some distance within which the gesture detection can occur. While determining the precise measures for describing an interaction zone of an input device is a topic for future work, we are currently investigating ways to define the interaction zone in terms of the physical and temporal aspects of target acquisition by considering potential variations on Fitts Law to address three-dimensional space. But for this framework, a rating of 0-1 can sufficiently describe the interaction zone. While devices such as remote controls and mobile phones are considered to have a 0 interaction zone, we introduce an additional parameter that addresses the mobility of a device within the framework.

Mobility. The mobility parameter for an input device is rated as 0 if it is not mobile, and 1 if it is. For example, a fixed camera has a 0 rating for mobility since it is typically held stationary throughout the interaction, whereas a PDA would receive a rating of 1, since it can be used with limited restrictions on its mobility. Other mobile devices, such as a wireless keyboard, or a remote control, which have constraints on their mobility receive a rating of .5 to indicate that there is mobility however it is restricted to being able to send signals to a stationary device.

5.3.4.2 System performance

A second element used to describe the enabling technologies refers to various performance measures associated with the gesture recognition system. This parameter does not refer to the input device, but considers the processor that enables gesture recognition with respect to a specific system. Performance parameters can be acquired from the software developers, system manufacturers or through running trials to determine the values of each. We use the following parameters used to measure performance.

System Accuracy. The accuracy of a recognition system can be dependent on external factors imposed from the interaction environment as well as on the specific ratings associated with the system. Perceptual input devices pose the greatest level of difficulty when attempting to measure recognition accuracy, since computer vision processing is often sensitive to changes in lighting, which can effect the overall accuracy of a system. When considering system accuracy, we can also factor in inconsistencies in user performance or in the type of gesture used in the interaction, however we maintain that a general rating of accuracy level as either low, medium or high will be sufficient for most scenarios. In cases where more in depth comparisons or analysis of systems is required, further assessments of accuracy can be used, addressing the quality of the recognition in terms of recognition sensitivity, or environmental changes as required. General measures of accuracy can be obtained for different systems easily if we assess accuracy under ideal conditions.

System response time. Typically, we would consider the speed of response for an interaction as an overall measure of the time taken to complete a task using a gesture. However, we have identified three separate stages of gesture interactions that can influence the speed of the interaction:

1. Input Stage: The first stage represents the time between when a user begins to gesture, and when the system acknowledges it as input.
2. Recognition Stage: The second stage represents the amount of time it takes for the system to process a gesture, or to complete the recognition process.
3. Response Stage: The third stage occurs while the system completes the intended interaction or task indicated by a gesture.

Although referencing all three stages of response time is unrealistic for many types of interactions, as in interactions where the response times are insignificant to the user, various circumstances may require individual stages to be considered. However for most situations, it is sufficient to consider system response speed as low, medium or high,

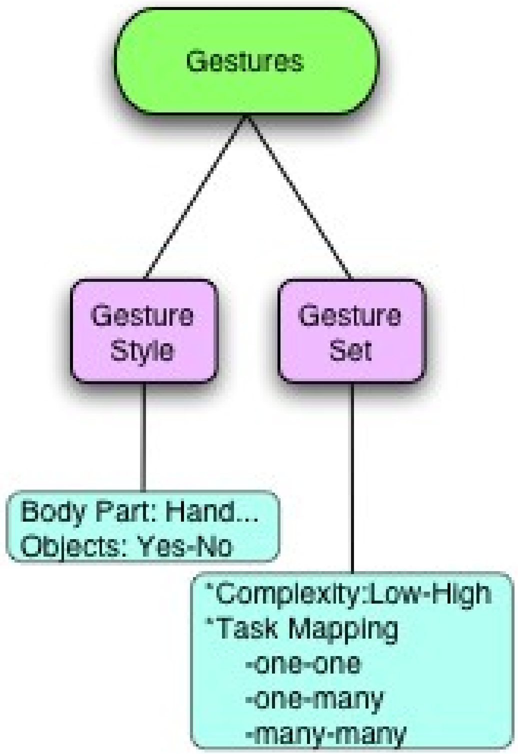

FIGURE 5.4: The diagram presents the elements of the framework considered under the category of gestures.

based on rating the actual system in use, and combining all three stages into the single parameter.

5.3.5 Gestures

The gestures category of the framework refers to the actual physical movements that the user performs to create a gesture. We do not consider any references to technology within this category, and focus strictly on the style of the gesture, the physical requirements required by the user, and a description of the set gestures used for the interaction. A diagram of this section of the framework is show in Figure 5.4.

5.3.5.1 Gesture style

Gesture styles currently considered for this framework are described in the classification as semaphoric, manipulative, deictic, gesticulation or sign language gestures, and discussed in Chapter 2. In this element of the framework, we identify the style or styles of gestures used within the interaction, described using the following parameters.

Body Parts. Each form of gesture has an associated body part that is responsible for the execution of a gesture. While gestures can be performed using the face, eyes, hands, fingers, full body, head and feet or any combination of body parts, there are some interactions in which the user must manipulate passive devices, objects or markers to assist in the recognition process. However, even when there is an external object used for the interaction, we can also specify the body part within this parameter as that which controls the object, such as using the hand to hold a stylus for example. In this case, we consider the object parameter, discussed next.

Objects. Objects in the context of this framework are those which act as an intermediary device that only assists in the recognition of a gesture, but is not responsible for the recognition processing. Objects are considered as markers or aids that can assist in the recognition of gestures, such as coloured objects or a stylus or pen. We use the value 0 to indicate that no object is used, or 1 to indicate that an object is used, and include the name or description of the object.

5.3.5.2 Gesture sets

We refer to the individual gestures included in the set based on the following parameters to provide a characterisation of the gesture set used in the interaction.

Task mappings. The framework considers the mappings of gestures are mapped onto their associated tasks: Mappings include one-one, one-many, many-many or many-one gestures to tasks, and is determined according to the specific interaction scenario. For example, a one-one mapping would assign a stop gesture strictly to one command, while a one to many gesture assigns the stop gesture to multiple commands when the computer is in different states. An example of a many to one mapping uses several gestures to control only one command, and a many to many mapping would consider using several gestures to control different tasks based on system states or contexts. This parameter is determined or set based on the specific characteristics of the interactions considered within the framework.

Gesture set complexity. Complexity of the gesture set is determined based on two factors: the number of gestures in a set and the physical complexity involved in performing a gesture. This parameter is set at values of low-high, indicating the level of complexity determined for the interaction. For example, a set of 10 gestures could impose a medium or high level of complexity on the interaction by requiring users to first memorise the set, while a single gesture set would likely be low in complexity. Complicated gestures that require the user to perform awkward hand movements for example could also lead to a high complexity for the gesture set. With sets that contain a variety

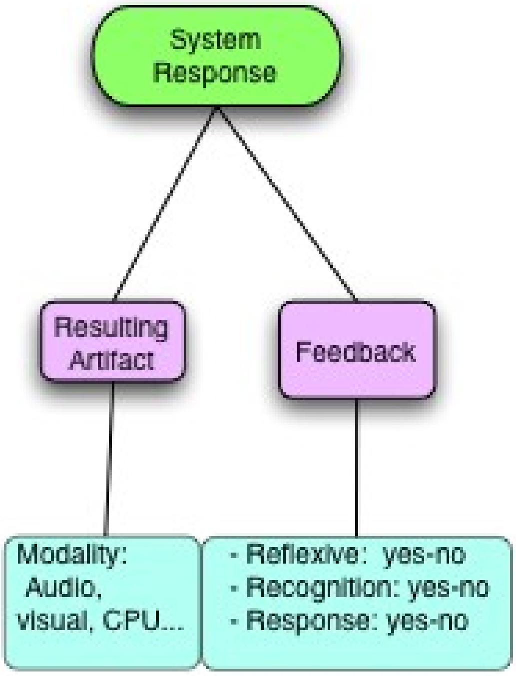

FIGURE 5.5: The diagram shows an overview of the elements considered within the category System Response of the framework.

of complex and simple gestures, we could consider the overall level of complexity as the average level of complexity for each gesture, however this is dependent on understanding the individual gesture sets.

5.3.6 System Response

In the framework, system response refers to the end result, or the outcome of a gesture interaction. It can be expressed in terms of the modality on which the outcome is expressed, presented, or designated as for further processing. A more compelling description considers the system response as an artifact that is the intended outcome of an interaction, as described using the task-artifact cycle (Carroll & Rosson, 1992), however for the framework, this reference is limited to artifacts that can be articulated in terms of what the user receives in the form of perceptual stimuli or behavioural responses resulting from the gesture. We next discuss the parameters for system response within the framework, shown in Figure 5.5.

5.3.6.1 Modality

As with the enabling technologies, the framework seeks to identify the modality in terms of a high-level description of the output device or system response that will present the resulting artifact of a completed task to a user. The high-level description of the modality of the system response includes audio, visual or a command based (CPU) responses, discussed in Chapter 2, however a more detailed description, where specific instances of a device can be expressed in terms of its technical specifications to enable a finer grained comparison between different devices. While this approach is only necessary when using the framework for informing detailed comparisons and analysis.

5.3.6.2 Feedback

An important part of the system response is the availability of some form of feedback to indicate that a response has occurred, that one of the input stages has been completed and the resulting outcome. Our research identified three stages of feedback that correspond to three stages of system response. While it may not be beneficial to present all levels of feedback in all interactions, their inclusion in the framework is to enable a more precise analysis when necessary, and their inclusion in the system can be indicated using a setting of 0 (not present) or 1 (present) for each type of feedback.

Reflexive Feedback. Reflexive feedback is the first stage of potential system notification to indicate the state of the input while performing a gesture. This level of feedback would provide users, who show a tendency to look up at the input device during training periods, or once an error has occurred (see Chapter 4) with information about the state of the processor. However, this level of feedback is most useful when using perceptual input devices such cameras or remote sensors to recognise gestures. While we hypothesise that this mechanism can improve user's overall performance and satisfaction when gesturing, we have conducted an experiment to investigate the efficacy of this feedback mechanism, which is discussed in Chapter 6.

Recognition Feedback. Recognition feedback occurs at the end of the processing stage, and can provide users with information about which gesture was recognised. This is included for completeness and when there are many different gestures used within a set, this may provide users with a mechanism to determine which gesture has been recognised, similar to a pop-up window intended for user verification of a selected command for example.

Response Feedback. The final stages of feedback occurs during the response stage of interaction, and provides a notification to the user to indicate that the task is complete. We consider this as traditional feedback, often resulting in an audio or visual signal to indicate the resulting outcome of the interaction, or is indicated by the accomplished task response.

5.3.7 Summary

In this section, we presented the main elements, and associated parameters, proposed within the structure of the framework. For each parameter, we introduced values that can be used to set or measure the performance of a gesture system to guide research, design, or evaluations. For example, when designing gesture interactions, the parameters can be set to reflect the individual requirements or goals that are to be met in the design, or used to describe and compare existing systems. In addition, future research towards the framework may involve making changes or adding parameters to the current structure in order to increase its validity or to reflect changes in the domain. However this is the intended goal of function of a theoretical framework developing using the GT approach.

5.4 Chapter Summary

In this chapter, we presented the approach, methodology and details of our framework for researching and designing gestures. Through repeated applications of grounded theory, we developed the framework to guide research and design of gesture interactions, and to provide a structure by which past and future knowledge can be understood. Although the framework, as presented in the next section has already undergone changes based on new research, this is its intended use and one of the key strengths in its purpose as a bridge for narrowing the gap that exists in relating user experience to engineering requirements. In the next chapter, we present three experiments that were conducted to investigate several parameters and categories of the framework, and discuss how we incorporated these results to validate the framework.

Chapter 6

Verifying and Extending The Framework

"An expert is a person who has made all the mistakes that can be made in a very narrow field."

Niels Bohr

6.1 Introduction

In this chapter, we discuss several applications for the framework, demonstrating its use as a tool for guiding research and design, and providing examples of how it can be used in different scenarios. We also present three experiments that were informed by the framework. The first experiment explores the predictive capacity of the framework, while the second uncovers results about an existing parameter. The third experiment investigates a new application domain to contribute additional propositions about parameters in the gesture category. I addition, we provide some examples where we use the framework to build relationships between the different parameters and their application, supporting a practical understanding of the interactions based on these relationships.

6.2 Examples of Applying the Framework

One of the key contributions of the frameworks is its capacity to enable relationships to be formed for the individual elements of the framework. Each parameter setting of the framework will have an effect on the others, and the relationships that are formed based on those settings supports a more informed process. There are two levels of relationships

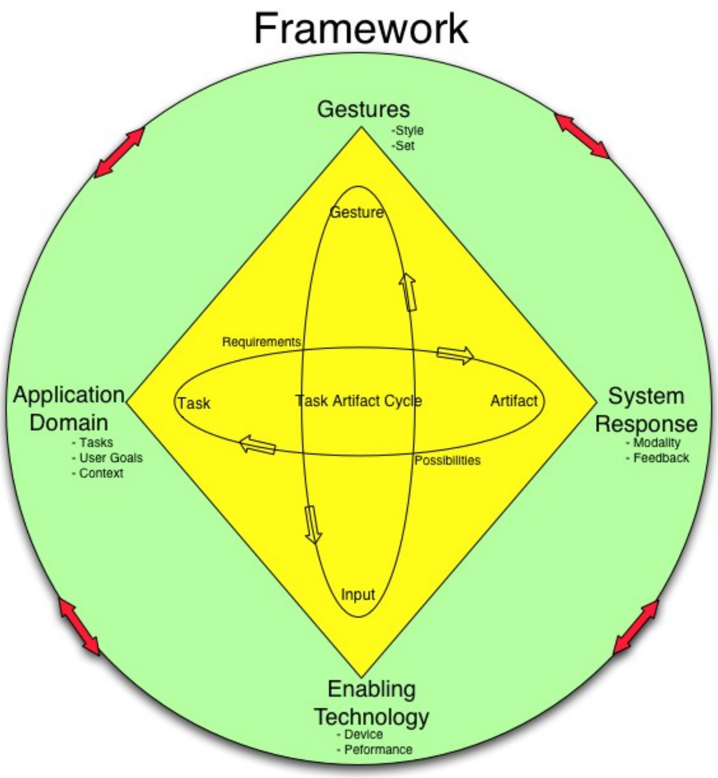

Figure 6.1: The diagram shows the fundamental relationship structure that can be drawn from applying the framework to individual systems for specific applications. The outer circle shows how we would approach relationships between element of an individual system. The inner circle represents different theories in HCI, and demonstrates how they can be incorporated into the framework for more general applications to designing gesture systems.

building that the framework supports; a specific level, where individual systems and the relationships between the parameter settings can inform design, and a general level, where existing HCI theories and methods can be incorporated into the framework.

Specific relationships. In Figure 6.1, the outer circle represents the process of examining each parameter within the categories with respect to the other, and forming an understanding about the different effects that can result. For example, we can look at the scenario that was used in the experiment presented in Chapter 4, and build a table to specify the different relationships that can be drawn between the variables considered in the experiment. This is demonstrated in Table 6.1 and discussed next.

General relationships. Figure 6.1 also allows for broader theories used in HCI to be applied to the design of gesture interactions within the structure of the framework. For example, the inner circle of Figure 6.1 suggests that the task artifact cycle, as introduced by Carroll & Rosson (1992), can be incorporated into the design of gesture systems under the categories of application domain to represent tasks, and system response to represent artifacts. Thus, the existing methods and approaches within the task artifact cycle can be incorporated into the design of a gesture system under this framework. Additional theories can be developed within the structure of the framework as well. The cycle that is shown in Figure 6.1 that runs between gesture and input could potentially be extended to form a similar theory as the task artifact cycle, only with reference to the gesture-input cycle. We do not discuss this further, but suggest that this would be a relevant direction for future research.

6.2.1 Assessing System Performance

The framework can be applied as a tool to inform on the potential effects of using different settings of the parameters in a system design. In addition, systems that are under design consideration can be evaluated using the values provided in this framework to rate the design before it is implemented. Table 6.1 demonstrates how metrics that describe the parameters can be used to provide comparison ratings of different system implementation. Metrics that are used to measure values of parameters can be set and cross referenced using a table format, to determine what an optimal configuration would be based on the information provided for those parameters. For example, in Table 6.1, we use four parameters to evaluate a potential gesture interaction system design (error rates, input mode, system response and task characteristics). By cross referencing each of the parameters against the others, we can determining a cumulative rating, based on the results of those combined parameters and settings, that can be used to predict the performance of that hypothetical system configuration. We can apply the information from the table in the following manner:

1. Select a parameter and setting from the row down the left side of the table (error rate = high).
2. Look along the columns on the top of the table, and locate the value for the first parameter and find its corresponding value for the appropriate value in the system response column (system response = fast).
3. Find the value to determine the rating for this particular configuration (med).

This process can be performed for all the parameters and their settings, and then assessed based on the collective ratings. In the above example, we see that a system with high error rates, and fast system response can at best provide mid-range interaction benefits. If we repeat the process and also consider task criticality and error rate, we get an additional rating of medium. We can then combine our collective ratings to determine how the system affects the interaction, making changes to reflect our design or goals. We can extend this table to incorporate the features of the interaction that designers are

		Error Rates		Alternate input		System Response		Task Critical	
		Low	High	Close	Far	Slow	Fast	Yes	No
Error Rate	Low	-	-	Med	High	Med	High	Med	High
	High	-	-	Low	Med	Low	Med	Low	Med
Alter Input	Close	Med	Low	-	-	Low	Med	Low	Med
	Far	High	Med	-	-	Med	High	Med	High
Sys Resp.	Slow	Low	Low	Low	Med	-	-	Low	Med
	Fast	High	Med	Med	High	-	-	Med	High
Task CRIT.	Yes	Low	Low	Low	Med	Low	Med	-	-
	No	High	Med	Med	High	Med	High	-	-

Table 6.1: This table presents several task characteristics and ratings to suggest their appropriateness for use in the given scenarios in relation to the other parameters that were evaluated in the experiment conducted in Chapter 4. The ratings are based on user feedback and results from the experiment, where we set low, medium, or high values based on the effects of the different settings for the variables.

considering, and use the resulting values from the framework to build the table. This is only one approach to using the framework, others can be adapted to a specific project or approach as required.

A new perspective for old research We propose that the framework can enable existing research to be incorporated into its current structure. For example, if we consider previous research by Brewster et al. (2003), who investigated gestures for mobile computing, we can apply the framework to impose a structure onto the existing system, and begin to understand its relationships with similar systems. To demonstrate, we demonstrate how we would apply the framework to describe Brewster work as follows:

- Gestures: Style:Semaphores, object:none, Set:Unspecified
- Application Domain: Context:mobile, multitasking. Tasks:non-critical
- Enabling technology: Touch Screen, Interaction zone:0. Mobility:1
- System response: Audio output, visual display (PDA), audio recognition feedback on,

Though we have used a reduced set of parameters to describe the system, we can chose to provide a much more in-depth analysis of the system provided that we have access to the information. Upon examination of the results obtained in Brewster's study, we see that here, semaphoric gestures reduce the level of distraction during while multitasking, and support eyes-free interactions. With this information in place, we now turn to a second application of the framework.

6.2.2 An old perspective for new research

If we again look at Brewster's work, we learn that touch gestures offer similar benefits to vision, however touch interactions provide greater accuracy than computer vision, and can potentially recognise a greater number of gestures. While it is not clear if we can always transfer the results from one interaction scenario to another, in this case, we see that one of the main benefits of using semaphoric gestures applies across interaction domains and enabling technologies. Of course, additional factors should be considered within the context of the interaction, and there are additional features that were not discussed in the research, however we do gain a general sense of the ability to view different systems within the framework parameters. To apply the framework, we consider our hypothetical system described in Chapter 4, which has a similar arrangement to Brewster's system:

- Gestures: Style:Semaphores, object:coloured objects. Set:5 gestures
- Application Domain: Context:ubiquitous, multitasking. Tasks: non-critical
- Enabling technology: Touch Screen, Interaction zone:1. Mobility:1
- System response: Audio output, visual display, recognition feedback: audio

If we want to improve our system, and enable a greater set of gestures, then we know that touch also provides minimal distraction, and can consider upgrading our system along these lines. We would also have to consider the trade-off of the interaction zone of vision for the for accuracy of touch, however this example demonstrates how we can make specific improvements and changes to systems based on comparisons with similar systems.

6.2.2.1 Framework Validation, Extension and Directions

An important contribution of the framework are its ability to proved a structure from which to inspire new directions for research. Within specific application domains, we can extend our research to consider interactions with individual systems. For example, while part of our research focus is on secondary task interactions using gestures, we noted that notification systems are also concerned with reducing distraction to tasks during multitasking situations, with a focus on the attention-utility theme to reduce the level of attention required from users while increasing the utility of that interaction McCrickard

& Chewar (2003). If we consider the benefits of using semaphoric gestures to reduce distraction, we can extend this concept and investigate if gestures can provide additional benefits to notification system interactions. To this end, the University of Southampton gesture researchers and notification system researchers at Virginia Tech combined to conduct an experiment to test this hypothesis. This study served two purposes: to verify predictions made using the framework about semaphoric gestures and distraction, and test if alternative interaction modes could affect notification system interactions. We discuss this experiment in section 6.3. We also present an experiment to test propositions about reflexive feedback and add knowledge about this parameter, in section 6.4, and finally, we discuss new directions for the framework in the application domain of CSCW and details of a formative study we conducted to investigate strategies for group gesture interactions in section 3.1.

6.3 Validating The Framework: Gestures and the Attention Utility Theme

This work was undertaken in collaboration with researchers at Virginia Tech, it is included as a demonstration of how we can validate the framework as a predictive tool for interaction design, and to demonstrate how it informed new research directions for notification system interactions, inspiring a study to compare affects of different input modes.

6.3.1 Related Work

The increasing use and pervasiveness of computer systems in command and control environments often result in people managing multiple information streams and engaging in several tasks simultaneously as demonstrated by Blandford & Wong (2004) and Williams (2000). In spite of the inherent problems with the interruptions these systems can cause, they are often vital aspects of the overall goals people are accomplishing. For example, large displays have been integrated into office and classroom environments to allow coworkers to maintain awareness of each others tasks and schedules while engaged in a primary work task (Ganoe et al., 2003; McFarlane, 2002). However, while ubiquitous systems can assist in managing multiple streams of information and supporting communication and group activities, users must also manage their attentional resources to glean the benefits of such systems. Recent work by Ou et al. (2005) identifies the value of efficiently directing users visual attention for task interactions, however alternative input modes may also provide a means of reducing the level of attention required to complete a secondary task.

Managing attention. Research in cognitive psychology has uncovered a number of limitations in human cognition with respect to memory and attention processes for example (Cowan et al., 2005; Peterson & Peterson, 1959). Attention research in particular is becoming more relevant as people increasingly use computers to engage in multiple tasks simultaneously. The key problem to address is the disruption caused to a current task when an interruption from a different system occurs. For instance, Czerwinski et al. (2000) observed the harmful effects of instant messaging notifications to the performance of a list evaluation task and the difficulty of switching between numerous tasks in Czerwinski et al. (2004). In addition, McFarlane (2002) has noted the relative lack of design guidance in terms of managing interruptions among this multitude of devices, and presents the results of an empirical investigation comparing the effectiveness of multiple design solutions to coordinate interruptions in a multitasking situation.

Attention-utility trade-offs. One way to mitigate the problems caused by interruptions is to design systems to be aware of users current context of use and to deliver notifications at times and in ways to minimise disruption. Horvitz (1999) have studied attentional user interfaces that treat attention as a scarce resource and use attentional cues to maximise value to users through a mix of automated services and informed notifications. This tension between maximising utility to users and managing their attention is also a key theme of McCrickard et al. (2003a)'s work in attention and system design. McCrickard deals with the development of notification systems —systems that deliver current, important information in a variety of platforms and modes without excessively disrupting users from their primary task. Notification systems are ideally suited for multitasking, computing situations that often arise in ubiquitous computing environments because their design is motivated by the need to accurately support attention allocation between tasks while simultaneously providing utility through access to information. (McCrickard et al., 2003b) have identified three critical parameters derived from the attention-utility theme —interruption, reaction and comprehension (IRC) —to guide the development of notification systems. These critical parameters are based on the work of Newman and serve as guides and benchmarks for developing notification systems (Newman, 1997; Chewar et al., 2004; Czerwinski et al., 2000). For example, a designer might determine that an alarm should have high interruption so it is more likely to attract the user's attention, support high reaction so a user can respond to the notification quickly and support low comprehension since alarms do not require users to understand and remember details about alarm events.

Multimodal interactions in ubiquitous computing The emergence of ubiquitous computing systems has motivated the development of a number of novel input methods that support richer interactions between people and systems including touch screens, voice control systems and gesture-recognition systems. However, different interaction techniques may not require the same levels of cognitive and physical attention from users, affording interaction characteristics that only provide benefits in appropriate contexts. These input methods are designed to support more intuitive, expressive means of interacting within environments that would be difficult or impossible with traditional desktop input modes such as a mouse and keyboard (Abowd, 1999; Oviatt et al., 2000; Chen et al., 2005). However, understanding when specific characteristics of input devices are most suitable can assist in better managing user attention and minimising disruption from primary tasks. For example, Anderson et al. (2002) reports that in-vehicle navigation systems may utilise voice-recognition technology to minimise distraction to drivers, while Thayer & Steenkiste (2003) demonstrate that touch-based interactions with mobile devices can support less disruptive eyes-free computing ideal for pervasive and social computing environments. Similarly, gestures may also provide a natural way to interact with computer systems at a distance; in particular, semaphoric gestures, which involve specific hand motions and configurations to communicate symbols, discussed in Chapter 3 may be ideal for interaction with secondary tasks supported by notification systems. But while it is important to understand the efficacy of different interaction modes and techniques, it is important to ensure that these findings can be readily applied to similar scenarios in order to transfer knowledge about the characteristics of the different interaction modes. We next discuss the use of an interaction model to inform the design and evaluation of a ubiquitous command and control computing environment.

6.3.2 Interaction Models

Beaudouin-Lafon (2004) argues that a shift from designing interfaces to designing interactions can significantly improve the way we approach system design. In our previous study, we proposed an interaction model for the design and evaluation of gesture based interactions, discussed in Chapter 4. The model considered three categories to evaluate interactions —interaction context, system performance, and user goals —and to guide our study on user tolerance for gesture recognition errors. We extend the interaction model to inform the design and evaluation of a command and control environment in this study. We investigate the efficacy of gesture and touch interactions while addressing the specific characteristics of our scenario and discuss these next.

6.3.2.1 Interaction Context

We define the interaction context for this research as notification system interactions within a command and control environment. In this scenario, users interact with a large screen display while responding to notifications that require secondary task interactions. Within this context, we consider two different notification system designs that are discussed next.

Multiple input devices and Notification Systems. This study considers the two interaction techniques under both high and low interruption settings within a notification system, and their ability to support reaction and comprehension goals across the different interruption levels. A high interruption system would correspond to critical systems that may require immediate attention such as a core-temperature monitor in a power plant while low interruption systems correspond to less critical tasks such as monitoring news or stock quotes while writing a report.

6.3.2.2 System Performance

We define system performance within the interaction model in terms of the individual input devices and how they can support the users in performing the secondary tasks.

Comparing interaction techniques. Through this work, we are beginning to explore characteristics of multimodal input interaction spaces for notification systems within the context of command and control environments. This work looks specifically at two promising interaction methods for notification systems: gestures and touch-based interfaces. Semaphoric gestures and touch were chosen as exemplar interaction techniques within the area of ubiquitous multitasking systems that we are focusing on. Both techniques employ similar cognitive and physical resources in terms of the potential benefits of exploiting people's pre-existing social, environmental and physical experiences with the world. In addition, both input modes can enable users to effectively manage their attention when working on multiple tasks in a command and control environment however, we hope to highlight specific performance benefits and issues unique to each interaction method. Touch-based interfaces support more accurate interactions but require users to be physically close to the system to use it. Semaphoric gesture-based interfaces may be less intuitive to use because they require users to remember specific motions, but they allow users to interact with systems at a distance.

6.3.2.3 Users Goals

Our work considers user goals and the attention-utility theme, where the multitasking situation requires that users maintain focus on a primary task, while ensuring that secondary tasks are correctly completed.

6.3.3 Experiment

We created a dual-task situation to simulate a command and control environment within the interaction model applied to this experiment. The primary task was a search task, requiring participants to locate a specific area of a satellite photo displayed at a resolution of 1024x768 on a 136cm x 101cm SmartBoard large-screen touch display within a specified time limit. The secondary task required participants to monitor a notification system animation presented on a peripheral display and respond when they received an alert using gestures or touch.

Approach. We investigated different settings of IRC values for notification system interactions to test how input modes can effect users and to determine their role in attention-utility trade-offs for designing multimodal command and control environments. We made the following hypotheses:

1. Interruption: Semaphoric gestures would support less interruptive interactions with notification systems, resulting in higher primary task performance.
2. Reaction: Semaphoric gestures would support more efficient reaction to both high and low interruption notifications, resulting in higher secondary task performance.
3. Comprehension: Users would experience similar comprehension levels across the different notification systems using either touch or gesture interactions resulting in similar success rates for secondary tasks.
4. User satisfaction: Semaphoric gestures would be as intuitive and easy to use as touch-screens to interact with notifications, resulting in higher subjective ratings.
5. Efficiency: Semaphoric gestures could better optimise attention-utility trade-offs, thus making them the preferred interaction mode for their perceived benefits.

We simulated user interactions on the peripheral display using the Wizard of Oz (WoZ) methodology, and provided two levels of interruption, low and high. The wizard used a wireless keyboard to control the peripheral display based on the participant's actions (gesture or touch). We chose the WoZ methodology rather than a working gesture recognition system to prevent confounding data resulting from recognition errors. Participants used single right-handed gestures or were required to touch the required area of the screen for the secondary task interaction. Participants were free to use either hand for the primary search task.

6.3.4 Experimental Design

The experiment was a full factorial, mixed-model, repeated measures design with two interaction modes (gesture and touch) as the within-participant factor, and notification system interruption level (high or low) as the between-participant factor. The interaction modes were counter-balanced within participants. For the low interruption condition,

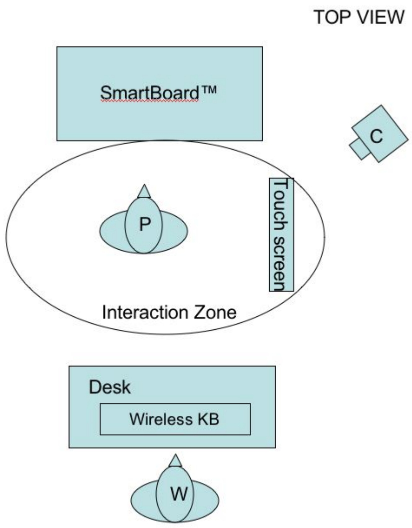

FIGURE 6.2: Overhead view of experiment layout with [c]amera, [p]articipant, [w]izard, kb = keyboard.

FIGURE 6.3: A user engaged in focal search task (left) in the presence of a secondary notification task (right).

visual notification was used, and in the high interruption condition, both an audio and visual notification were presented to the participants. We tested 4 conditions, coded according to the interruption level of the notification (0=low, 1=high) and counterbalancing (G=gestures seen first, T=Touch screen seen first). The four conditions are thus 0G, 0T, 1G, and 1T. In this paper, we define interaction zone as the area in which a user can conduct purposeful interactions with the system; as shown in Figure 6.2. For the gestures, the interaction zone coincides with the camera's field of view, and with the touch screen, the interaction zone is the area on the screen that the user must touch to issue the command.

During the course of the primary search task, participants were presented with a series of notifications, occurring at 20-25 second intervals. In the low interruption mode, notifications consisted of a visual strobe lasting 0.333 seconds each second (1/3 duty cycle). In the high interruption condition, the notification was issued as a visual strobe (1/2 duty cycle) with an initial beep lasting 1 second, available from the Java 2, Standard Edition, v 1.4.2 sample code (TicTacToe applet). Sound was issued from speakers positioned near the secondary display for a spatially appropriate cue. Figure 6.3 shows a participant working on the primary task during the experiment. Our prototypical gesture set consisted of left-ward, vertical, and right-ward hand motions. We chose gestures that were easy to perform and had straightforward mapping to concepts (left, middle, right). We wished to minimise the cognitive effort required to perform the correct semaphore in response to notifications and to ensure that the touch and gesture interactions posed similar physical requirements. In this case, the gesture path encoded the relative positions of the coloured bars present in the notification display. The gestures were understood to serve the function of acknowledgement as well as having specificity for the particular notification issued. We define our dependent and independent variables next.

Independent variables. The independent variable tested for the within-participant condition was interaction mode. Each participant used both the touch screen and the gesture interaction. The between-participant variable was the notification system interruption levels for secondary tasks: Low interruption provided visual only notifications and high interruption used both visual and audio notifications.

Dependent variables. To determine the measurements for participant reaction, comprehension, the degree of interruption to the primary task and the efficiency of semaphoric gesture vs. touch screen, we measured reaction time (reaction), success rate for responding to notifications (comprehension), the time taken to recover from the interruption to the primary task (recovery) and secondary and primary task times (efficiency), defined below. We also gathered subjective data on user preference and satisfaction ratings for each interaction mode using a post-evaluation questionnaire. Our experiment software recorded the times for each notification issued and the time that the primary tasks were resumed after completing the secondary task. Response times for notifications were logged using a key-press at the Wizard's station. We measured the relative ease of execution of gesture vs. touch based on significant differences in reaction and recovery times. To infer effects on attention, we calculated the search success rates and measured the search times (primary task). We also recorded the number of secondary tasks completed and the total time to perform them. A detailed definition of the dependent variables is given below:

- Reaction: The time taken to respond to a notification, measured from when the notification is issued to when the user reacts by touching or gesturing.

		Mode			
		Gestures		Touch Screen	
		Reaction	Recovery	Reaction	Recovery
		Mean	Mean	Mean	Mean
Condition	0G	2.984	1.868	3.238	1.921
	0T	2.362	1.999	2.286	2.282
	1G	2.410	2.082	2.424	1.914
	1T	2.103	2.018	2.281	2.256

Table 6.2: Mean times for reaction and recovery for gestures and touch screen.

- Recovery: The time taken for the participant to resume the primary task, measured from when the secondary task ends and the primary task resumes, representing the level of interruption caused to the primary task.
- Primary task success rate: The number of times an image was found during the focal search task.
- Secondary task success rate (comprehension): The number of times the participants were able to correctly respond to the alert within the allocated time frame.
- Primary and secondary task times: The total amount of time participants spent during the search tasks, and responding to the notifications.

6.3.5 Results

To check for significant differences in the results, we first ran a mixed-model repeated-measures analysis of variance (ANOVA) to test the effects of interaction mode within participants and condition (order in which the two modes were presented to participants and the interruption level) between-participants. A second multivariate analysis of variance (MANOVA) was run to examine the effects due to the independent variables of mode (gesture or touch), interruption level and counterbalancing in isolation of the four conditions. While our ANOVA showed no significant results for the within-participants factor of interaction mode, significant results were found in the differences between participants for primary and secondary tasks success rates, reaction time and secondary task time for the four conditions, discussed next.

Reaction and recovery time. Reaction time was significant in the ANOVA, with the mean reaction time for gestures faster in all but the 0T condition. This may be due to the more relaxed nature of the interactions within the low-interruption level conditions and possibly due to dealing with the novelty of the gestures in the second set of trials. The MANOVA revealed that an interaction effect existed between modes and

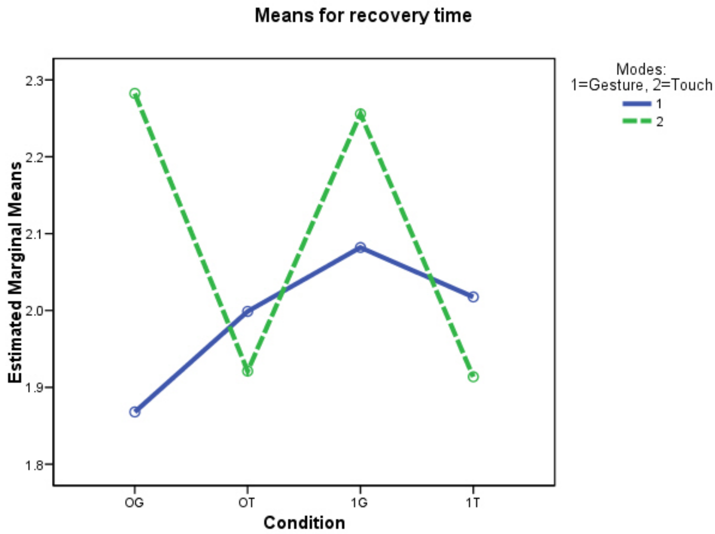

FIGURE 6.4: Estimated marginal means of recovery time for modes: gesture and touch. Recovery time when gestures were shown to participants first were higher. This is likely due to the novelty of the gestures. We see lower recover times when gestures are seen in the second set of trials indicating that there is a learning effect present for the trials.

counterbalancing, thus suggesting a learning effect was present. The MANOVA also revealed that interruption level yields a faster reaction time in a high interruption notification than for low interruption ($F_{(1,32)}$ =6.383, p<.05). While no interaction effect was shown between gestures and interruption level in the MANOVA, reaction times for gestures tended to be faster than those for touch screen (see Table 6.2). An interaction effect is shown for reaction time, with mode and order of presentation, suggesting that a learning effect positively affected reaction times for gestures in certain cases: In condition 0G, reaction times for touch actually increased, while in 1T reaction times for gesture decreased during the second trial, with 0T and 1G having similar reaction times ($F_{(1,32)}$ =9.583, p<.005) (see Figure 6.4). Our analysis did not show any significant differences in the recovery times, however, there is a definite trend towards faster recovery using gestures over the touch interaction, as seen in Table 6.2. This applies to all but the 1G condition, where gestures appear to lead to a slower recovery time than for the touch interaction. While the slower recovery time may be due to increased interruption level, and to the novelty of using gestures, the differences are not shown to be significant in this model.

Primary and secondary tasks. Significant results for interaction mode in the ANOVA for primary tasks in the 0G condition with the gesture interactions yielded a greater number of primary tasks completed ($F_{(1,8)}$ =6.733, p<.05). Secondary tasks were also significant in this model ($F_{(1,8)}$ =5.829, p<.05), however there were fewer secondary tasks completed for the gestures (Table 6.3). Significant results were found

		Mode			
		Gestures		Touch Screen	
		Primary Tasks	Secondary Tasks	Primary Tasks	Secondary Tasks
		Mean	Mean	Mean	Mean
Condition	0G	9	25	4	31
	0T	8	30	10	22
	1G	8	25	9	29
	1T	11	30	7	26

TABLE 6.3: Mean values for primary and secondary tasks completed for the two interaction modes used in the experiment.

in the 0T condition, however gestures lead to lower success rates for primary tasks ($F_{(1,8)}$ =6.897, p<.05) but a greater success rate for secondary tasks ($F_{(1,8)}$ =9.218, p<.05). The 1T condition suggest that gestures also support a higher success rate for primary tasks ($F_{(1,8)}$ =9.529, p<.05). No significant results were found for the 1G condition. The MANOVA results show that overall, completion of secondary tasks are significantly greater in the second set of trials ($F_{(1,32)}$ =14.286, p<.001) which may explain the difference in primary task success rates between the four conditions in terms of mode from the ANOVA. There is also a significant interaction effect present for primary task success rate ($F_{(1,32)}$ =16.711, p<.001) due to the factors interruption level and presentation order. In the first set of trials, there are more primary tasks completed for low interruption than for high interruption. However in the second set of trials, fewer primary tasks were completed in the high interruption condition than for low interruption. This suggests performance degradation occurred with higher interruption since it draws attention away from the primary task.

Subjective results. Despite the limited differences observed for the performance measures, participants showed an overall preference for gestures over the touch screen interaction (mean 7.15/10 for gesture). Participants rated gestures slightly higher than touch-screen for ease of interaction (mean of gestures=7.41; touch=7.22) but performance was rated as slightly lower (mean of gestures=8.52; touch=8.78). Ratings for ease of resuming the primary task were higher for gestures (mean gesture=7.11; touch=5.22), and lower for attention required for secondary task (mean gesture=5.37; touch=6.74), and distraction caused (mean gesture=4.67; touch=6.63), shown in Figure 6.5. When we examine these results, participant perceptions were that gestures required less attention in the low interruption condition than in the high interruption condition. Touch screen interaction was roughly equivalent for both high and low interruption conditions. Gesture and touch were both rated as less disruptive in the high interruption condition while resuming tasks in general was rated as easier in the low interruption condition.

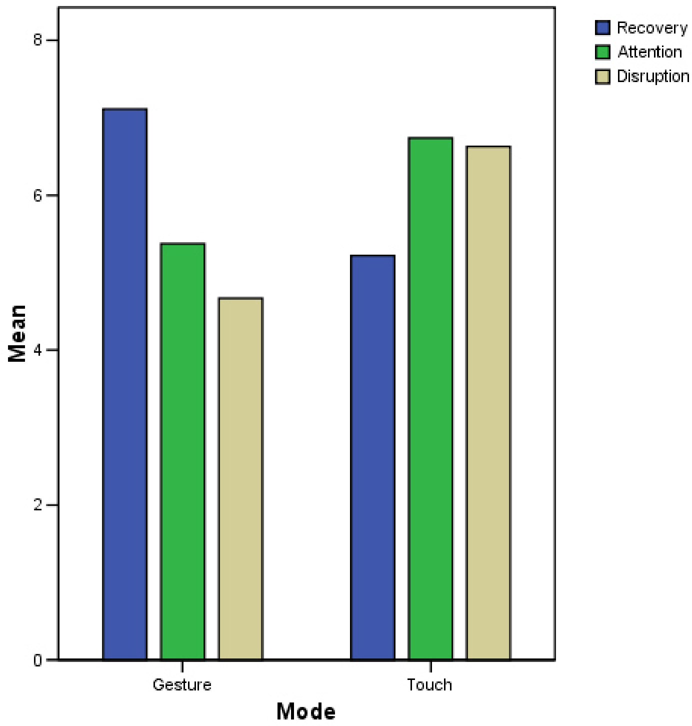

FIGURE 6.5: Summary of subjective ratings for interaction mode, gesture or touch screen.

6.3.6 Discussion

We now present several areas that the results of our study address, and details on how multimodal interactions relate to the attention-utility theme and how the results of the study validate the predictions of the framework.

Attention management. In the low interruption condition, where gestures were seen first, the number of primary tasks completed was significantly less for gestures than for the touch screen interaction. This suggests there is likely an element of difficulty in using gestures for the first time (most participants had little to no prior experience with gesture systems). This is also consistent with the reaction time for gestures being greater than the reaction time for the touch screen for the first set of trials. Based on observations, participants at first tended to use deliberate, slow gesture motions when reacting to notifications. However, by the second set of trials, participants had become accustomed to the interaction (i.e. accounting for warm-up time). During low interruption, we see a greater number of primary tasks completed in the second group of trials for gesture compared to touch screen. Semaphoric gestures provided a means of acting at a distance, which was useful for managing secondary tasks. When gesturing, users were able to maintain their focus the primary search task, and only had to glance at the secondary display before gesturing. In the touch condition, participants

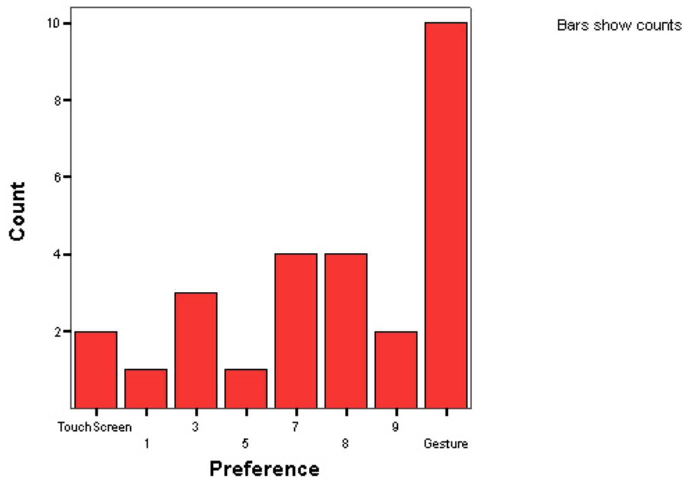

Figure 6.6: Users ratings for interaction mode preference (Touch-gesture).

had to devote more of their attention to the secondary display because they physically had to move to the display and touch a specific area of the screen. Our results show the most benefit for gesture occurs when attention draw is low. The benefit appeared to be masked in the high interruption group, as there was no significant difference in performance. We note that the higher interruption was an increased rate of visual strobe with an accompanying sound; as expected, this was sufficient to reduce the reaction times. However, the greater level of distraction meant that participants lost the benefit of gesture over touch-screen interaction. We provided a spatially appropriate cue, which caused eye gaze to be diverted to the notification task. This effect interfered with the eyes-free benefit of gestures. While our results show that the significant differences in performance are primarily affected by condition rather than interaction mode, we observe that semaphoric gestures support less interruptive interactions with notification systems than touch-based interactions for low interruption secondary tasks. Our first hypothesis is thus supported for non-critical (less interruptive) secondary tasks.

Interaction mode and utility. Overall results suggest that reaction times and performance measures in the gesture condition do not differ significantly from the touch screen condition. This suggests that our second hypothesis is not supported when the interaction space for gestures and touch are similar. Since our interaction scenario is based on using similar interaction zones for both gestures and touch screen, we observe that the increased area of the interaction zone possible with gestures would thus be more suitable for a pervasive or ubiquitous computing environment where interactions at a distance may be appropriate. As the surface available to a system becomes larger, the interaction zone scales also, resulting in larger spaces for interacting compared to

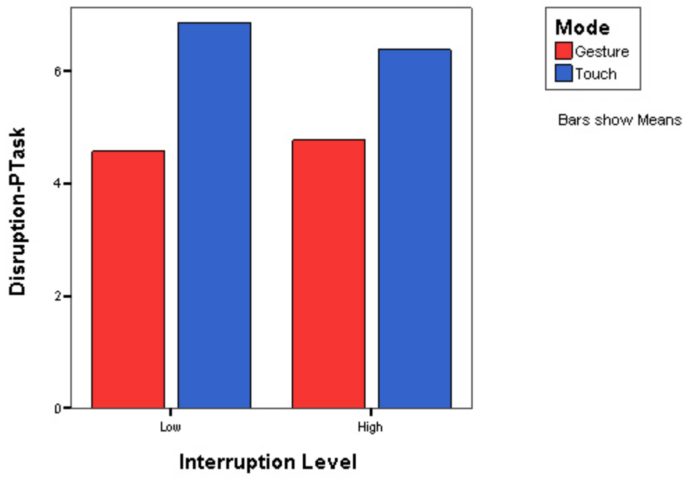

FIGURE 6.7: Subjective results showing disruption to primary task organised by interruption level conditions and mode.

what is typical of desktop applications. The performance benefits of semaphoric gestures over touch-based interactions would become clear as the interaction zone scales larger. The very low error rates in participant responses to notifications show that they had no problem understanding and responding to the notifications in both the touch and gesture conditions, showing that both are equally useful for performing relatively simple interactions that are often required from notification systems. However, if we also consider the subjective preference participants showed for the gestures over the touch screen, and overall primary task completion rates, we can deduce that gestures have the potential to improve utility (secondary task performance) while reducing the demands on visual attention.

Interaction mode and the attention-utility theme. We also varied the input mode in order to observe whether differential effects occurred in success rates of primary and secondary tasks. Results suggest that our third hypothesis is supported, as secondary task completion rates or comprehension levels were not effected by input mode, but by other factors such as familiarity with the task. Studying the impact of input mode on the user is an area that is just beginning to be explored in the design of notification systems. We selected touch screen as a conventional input in comparison to semaphoric gestures, a novel input method in this dual task scenario. Our hypothesis that semaphoric gestures can enable a more effective style of interaction that reduces distraction to a primary task was supported in the group with lower interruption. However, during high interruption using gestures had no additional benefit compared to the conventional touch interaction. Given that the results show differential effects depending on level of interruption, we observe that in a multi-tasking situation the choice of input mode can also have an impact on the user's reaction to secondary tasks. In particular, gestures for responding to notifications permitted a lower level of distraction, whereas touch screen often resulted in loss of context in the focal search task. We note that the difference between gesture and touch disappeared in the high interruption group. The user response to a notification requires an allocation of attention, which may vary depending on how they interacted with the notification system. Thus, semaphoric gestures permit less interruptive interaction when distraction is already low; in terms of reaction and recovery times, the efficiency of reaction for gesture is similar. When considering these results, we can conclude that our fourth hypothesis; that gestures are more efficient for managing attention-utility trade-offs is supported for non-critical notification tasks.

Interaction mode and user preference. Since gesture currently is a novel input mode for many users, this type of interaction can require a period of adjustment. This suggests that semaphoric gesture is not as natural an interaction technique and thus does not engage our preexisting cognitive and physical resources as effectively. However, we observed that gesture performed as well as touch-screen, and was subjectively the preferred mode of interaction for the majority of users. This shows that our fifth hypothesis that semaphoric gestures are as intuitive and easy to use as touch-screens to interact with notifications is only partially supported. Most participants experienced gestures as a less disruptive interaction, and it permitted easier resumption of their primary task. For gesture, several users reported an ability to search without feeling tied to the secondary display. The benefit of gesturing in-place avoids re-purposing ones hands to carry out a secondary task such as responding to a notification, and avoids changing ones physical position such that the current context in the focal search task is lost. Thus, after a period of adjustment, gesturing was seen as easier by the majority of participants.

Notification systems and multimodal interactions. Using the IRC framework, we were able to select two different styles of notification, knowing in advance there was a salient difference in the level of interruption. This allowed us to expect certain behaviour based on our design decision. In particular, we expected faster reaction times for higher interruption, without any impact on the performance in the primary task. We also expected gesture to be a better means of interaction in all cases based on the following design claim: Semaphoric gestures for secondary tasks are convenient for action at a distance and provides opportunity for eyes-free responses to notification (i.e. use of peripheral vision) but may require more time to execute compared to traditional input methods. In this study, varying the level of interruption allowed us to probe two areas of the IRC design space. Our results highlight the importance of addressing different input modalities and the usefulness of the IRC framework for designing notification systems. First, the level of interruption has a direct influence on the user's reaction as expected. Second, the input mode has an impact on reaction that must be considered as part of the design. We note that the area of interaction zones will continue to grow as computer systems and sensor networks provide increasingly aware spaces. According to work by Ho & Spence (2005), and depending on the application, maximising utility while minimising the cost to attention will require sensitivity to both endogenous (user-directed) and exogenous orienting mechanisms of attention. As systems tend to become increasingly pervasive in support of "off-the-desktop" computing, we expect that the choice of input mode will favour semaphoric gesture when lower distraction is desired; this will be made possible by supporting the user's ability to act at a distance. Our work shows how design goals, stated here in terms of the attention-utility theme and the IRC framework, can be related to effective choice of input modality for a specific class of ubiquitous systems.

6.3.7 Contributing to the Framework

Since we have evidence to suggest that gestures can reduce the level of distraction caused to a primary task during multitasking situations, the study in this section demonstrates how results made in previous research are supported by results in this experiment. While the inclusion of this study is as a verification of the framework, a more important issue is presented: where we can begin to consider the notion of alternative interaction modes as a factor in designing interactions.

In this study, gestures controlled a large screen display during a multitasking situation, in which a notification system was used to signal that a secondary task was to be completed. The study compared gestures to a touch screen interaction where the primary task was located on a large screen display, while the secondary task was presented on a smaller screen in the users peripheral vision. The important issue here is to demonstrate, on a category level, the predictive ability of the framework on our hypothesis that gestures can reduce distraction in multitasking situations, and to stress the novel perspective of considering different interaction techniques as a method of improving notification system designs.

Gestures, touch screens and notification systems. This study demonstrates the predictive nature of the framework when we consider interactions along similar categories and parameter settings. In addition, novel information gained through the study can be incorporated into the framework to increase its completeness. The study addresses the following aspects of the framework:

- Gestures: semaphoric, bare-hand, 3-gestures set, low cognitive and physical requirements.

- Application Domain: Ubiquitous computing; command and control environment; task: respond to alert issued using notification system, with low or high interruptions
- Enabling Technology: cameras or sensors, with WoZ approach to the camera.
- System Response: visual display, with audio feedback, no reflexive feedback.

Results from this study can contribute additional research directions within the framework. For example, one result of the study supports evidence that gestures, with their large interaction zones are less interruptive to primary tasks in multitasking situations during low interruptions levels of a notification system. While the concept of interaction zone is presented in our framework, there are currently no parameters in place to enable us to discuss an interaction zone in quantifiable terms. This will be discussed in Future work, presented in Chapter 7.

Results also suggest that participants are more comfortable using familiar interaction techniques such as the touch screen over gestures. This finding supports previous results indicating that gestures are less suitable when there is a direct input device available within reach of the user (Chapter 3). A third contribution suggests that gestures were found to be better than the touch interface at managing attention-utility trade-offs, in that they were less disruptive to an ongoing task, while supporting adequate reaction to notifications. This result supports findings presented in our previous research, further validating the framework. And finally, results suggest that there were similar reaction times when responding to notification using gestures or touch interactions, affording similar comprehension levels, and further validating results presented in our framework.

6.4 Investigating Parameters: Reflexive Feedback

Having identified the three types of feedback within the framework, we demonstrate new research conducted to further explore the feedback parameter of the framework. This study explores the notion of providing visual feedback to users when they are executing semaphoric gestures. We call this reflexive feedback, because the computer provides a view of what it is seeing in terms of the object being tracked. While the idea of reflexive feedback is not new, its use as a mechanism of feedback for gesture interactions with computer vision is. Based on our methodology for conducting research into our framework using quantitative methods to contribute measurable data to the parameters, we hypothesise that reflexive feedback will improve interactions with gestures by enabling users to better recover from errors, and avoid performing gestures out of the range of the camera used for input, and to provide a visual mechanism for self-tracking of gesture performance.

Unlike direct-input devices, where the visual feedback of a mouse for example, is represented on the screen, gesture technology does not provide an obvious mechanism to enable users to understand the system state during tracking stages. We coined the term "reflexive feedback" as a mechanism to provide users with the kinematic information necessary to increase their ability to perform gestures within the visual range of the camera and hypothesised that this would improve accuracy rates for recognition systems. We also predicted that users could achieve a greater level of mobility and flexibility in a ubiquitous computing scenario, where users are not always in the same location, where it would be difficult to maintain awareness of the interaction zone of the camera. While some form of visual feedback is typically built into gesture recognition systems (Nickel & Stiefelhagen, 2003; Turk, 2004), we have not located any work that investigates the effects that this has on the user experience during gesture interaction.

In this work, we describe an implementation of reflexive feedback, where the computer displays a visual representation of the image transformations that occur during tracking, to provide users with kinematic understanding and orientation of gestures during execution. In the next section, we describe the results of our pilot study, and the implications for the framework.

6.4.1 Exploring Reflexive Feedback

Reflexive feedback is one method of providing users with information about their location within the visual field of a camera while gesturing. Potential benefits of reflexive feedback were realised after observations during past experiments (see Chapter 3. We observed a common behaviour where participants looked up at the camera, primarily during the training stage or after an unexpected system response or false positive recognition. While it is not clear what they expected to learn from the camera, post-experiment interviews revealed that participants wanted to ensure the gestures were within range of the camera. This tendency to move eye gaze towards the camera during gesture interactions prompted us to provide a source of feedback to assist users and provide relevant information to increase the awareness of the recognition system.

Although the rationale behind this behaviour seems elementary, the factors that influence the errors that occur with computer-vision enabled gestures are often related to factors other than ensuring that the gestures are being performed within visual range of the camera. Thus, lighting changes, performance speed and range of gesturing can also influence the success rates of recognition. Given that this information is not possible to gauge from simply looking at the camera, we predicted that reflexive feedback could assist users with better understanding of the recognition problems that can occur during gesture interactions. Unlike direct-input devices (keyboard, mouse, remote control) where there is a tight feedback loop between the input and the output processes, there is a gap in feedback available for gesture systems, which is typically presented only after

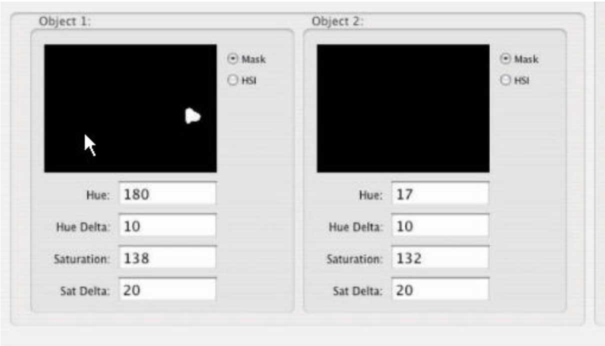

FIGURE 6.8: iGesture Reflexive Feedback Window. The white blob in the left window represents the purple object being tracked by the iGesture system.

the gesture is executed, bypassing any information to indicate why there was a failure. Reflexive feedback can provide pre-recognition information that we predict will enhance the interaction for the user. While computer-vision enabled gestures can enable eyes-free gestures, we consider the notion of reflexive feedback from the perspective of providing support to users when there is a screen available. However, there are many issues related to providing this feedback, including how this can be done where there are not visual displays available, discussed in section 6.4.5 of this chapter.

6.4.2 Reflexive Feedback and iGesture

The iGesture platform, described in Chapter 3 presents the reflexive feedback in a window visible on the iGesture interface. It displays the object that the camera is tracking as a white blob on a black background, shown in Figure 6.8. The window isolates the object that it is tracking from the background to provide a visualisation of what the computer is currently tracking, shown in Figure 6.8. This contrast enables users to quickly identify their movements on the feedback window while they are performing the gestures. As the user gestures in front of the camera, they view the feedback window and ensure that gestures are properly tracked. Although reflexive feedback can be implemented using different visualisation's, the iGesture system presents just one possible solution which is based on the transformation of the object that is being tracked.

6.4.3 Evaluating Reflexive Feedback

We wanted to determine if reflexive feedback offered any significant interaction improvements for gesture based interfaces. We hypothesise that reflexive feedback would improve the accuracy of the gesture recognition, increase users confidence and satisfaction, and be preferred over having no feedback window present. We next present a summary of the pilot study we conducted and our results.

Tests of Between-Subjects Effects

Source	Dependent Variable	Type III Sum of Squares	df	Mean Square	F	Sig.	Noncent. Parameter	Observed Power[a]
Model	False Negatives	1562.063[b]	2	781.031	33.333	.000	66.666	1.000
	Recognition Error	1192.063[c]	2	596.031	21.467	.000	42.935	1.000
RFWindow	False Negatives	1562.063	2	781.031	33.333	.000	66.666	1.000
	Recognition Error	1192.063	2	596.031	21.467	.000	42.935	1.000
Error	False Negatives	702.938	30	23.431				
	Recognition Error	832.938	30	27.765				
Total	False Negatives	2265.000	32					
	Recognition Error	2025.000	32					

a. Computed using alpha = .05

b. R Squared = .690 (Adjusted R Squared = .669)

c. R Squared = .589 (Adjusted R Squared = .561)

Table 6.4: The table shows the significant results of the MANOVA run on the independent variable feedback window, and the two dependent variables false negatives and system errors.

6.4.3.1 Experiment Design

We ran a pilot study to explore our hypothesis that reflexive feedback can improve gesture based interactions. The experiment was a within-participant, single factor, two level, counter-balanced design. 17 participants took part in the study, 10 from Virginia Tech, and 7 from University of Southampton. Participant performed two sets of trials, each consisting of 50 gestures using the iGesture platform: 10 sets of 5 gestures, stop, up, down, left and right movements across the camera. Each participant performed the gestures as the researcher called them out during the experiment. We measured the number of false positive responses and the incorrect recognition of gestures made by the system. The average response speed of the iGesture occurs in under 1 second of performing the gesture.

Each participant was given a pre-evaluation questionnaire, and then trained to use the gestures. Participants could practice until they were comfortable. There were a total of 100 gestures per experiment session, 50 for each of the two conditions (feedback window present or not). During the trials, the researchers noted the results of each gesture as a success, a false negative recognition or an incorrect recognition, in which case the gesture that was mistakenly recognised was recorded. In the case where a user performed an incorrect gesture, we recorded the intended gesture and the system's response. We also gathered qualitative information using observations and post-experiment questionnaires and interviews with each participant.

6.4.4 Results

We ran a multivariate analysis of variance on the data, removing one outlier case where there were complications with the trial, resulting in excessive errors in the feedback

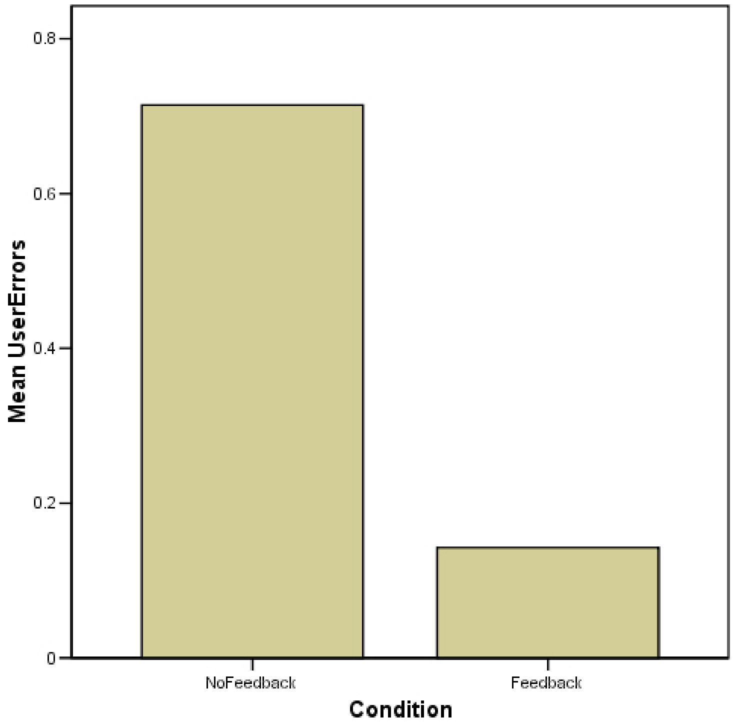

FIGURE 6.9: Bar graph shows the errors that users made when gesturing with and without the feedback window present.

window condition. Results showed significant differences in the error rates for false positive and incorrect recognition errors for the independent variable, reflexive feedback (see Figure 6.9 for user errors). Both false positive and system errors were significantly lower in the conditions where participants used the feedback window (false negatives: $F_{(2,32)}$=33.333, $p<.001$, system errors: $F_{(2,32)}$=210.467, $p<.001$)) as shown in the ANOVA table 6.4.

Subjective results. Results show that most participants preferred the feedback window however there were several participants who felt that the window was distracting. Since this was a controlled experiment, the set up would suggest that the interface is distracting, since the participants were sitting at the computer performing gestures as their primary task. This made looking at the feedback window a secondary task, and since there was no other visual stimuli or primary task for the user to focus on during the trials, it is natural to look at the screen. In addition, results presented in Chapters 3 and 4 indicated that users preferred not to use gestures when in a desktop scenario, and the user's comments in this experiment further support this result. However, the quantitative results support our hypothesis that the reflexive feedback window can provide more accurate interactions with perceptual style gestures, and users comments also demonstrate how the feedback window can improve gesture interactions in general. Although some of the participants did prefer the feedback window (see Figure 6.10), this

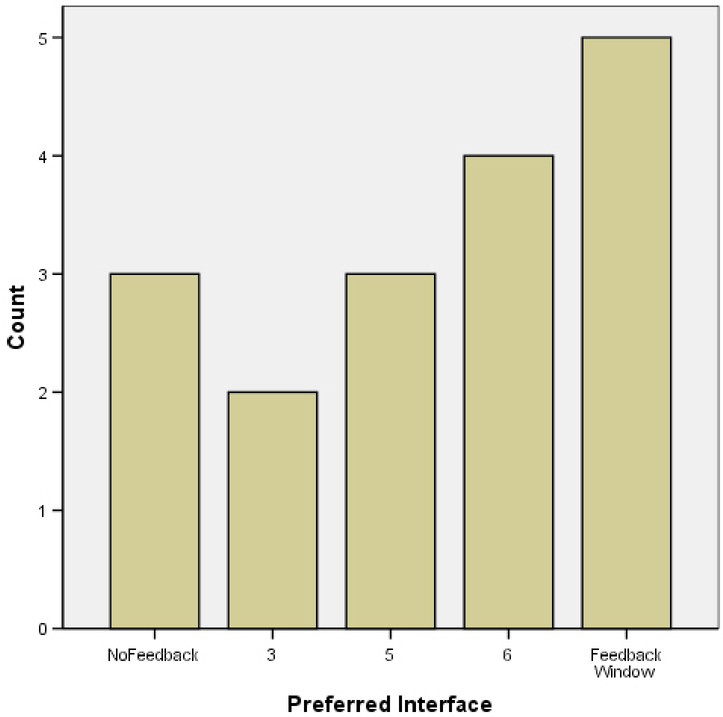

FIGURE 6.10: The graph shows the responses of participants for their preferred interface, with the feedback window on the left side of the graph, and without the feedback to the right.

may be a result of the interaction context. Experience suggests that gestures will not be chosen over a direct-input device when on is available, as in a desktop computing environment. Participants commented that the feedback window was distracting.

Qualitative results. We observed users during the interactions in both cases, noting that they continued to look up at the camera in most cases where a gesture was not recognised. This result supported observations from our previous study (see Chapter 4). From our interviews, we also noted that the specific location of the feedback window on the screen, which was only used to display the feedback window, was a factor that users indicated they would like to control. Some participants indicated that they would like to have the feedback window located directly under or beside the camera, while others suggested having the window located closer to them, and possibly away from the camera to enable viewing the window in a more convenient location.

6.4.5 Discussion

Results from this experiment support our hypothesis that the reflexive feedback window can improve user interactions with gesture based interfaces by reducing the number of errors and by providing kinematic feedback for the user during gesturing. Additional results showed that the participants were very pleased with the feedback window and found it to be very useful in assisting with the gestures during training, however many stated that in this situation, where they were sitting at a desktop computer, there would be problems in placing the feedback window in an unobtrusive location on a display. Participants also expressed concern about the value of using gestures in a desktop computing scenario, supporting previous work (see Chapters 3 and 4). While this was a pilot study, we acknowledge that a more realistic study within a real-world scenario could support a more practical approach to determining the usefulness of the window. However, the results of the study demonstrated that accuracy rates increased using the window and motivate future work on reflexive feedback. Some of the issues we considered for future direction within this area include exploring different visualisation's for the RF window: while we concentrated on a black and white representation of the tracking process, we could still learn more about providing the actual image, and its affect on the interaction. Additional issues arose for the placement of the feedback window, the representation of RF when visual displays are not available, and using different graphical representations for the tracking data.

6.4.6 Summary

Results of our experiment suggest that the presence of the reflexive feedback window does assist users in improving their performance on gesture, and provides a greater sense of understanding and confidence in the system while learning and performing gestures. Many users were distracted by the feedback window, but this may have been due to the context, since we know that desktop interactions with gestures are not ideal (see Chapter 4). Future research on reflexive feedback is intended to investigate different versions of the window, and ways in which the window can be used in a real-life ubiquitous computing scenario. This result contributes additional knowledge to the feedback parameter in the framework, and demonstrates how now research can improve the validity of the framework.

6.5 Group Interactions and Gestures

A third experiment contributes to the structure of the framework in the domain of CSCW interactions. We conducted a formative study to investigate group interactions with the iGesture system. 40 participants took place in this experiment, conducted in groups of 2,3 or 4 participants. The study was qualitative in nature, exploring different strategies for distinguishing gestures from different people. We discuss several approaches to enable group interactions that involved a variety of techniques. Our results on the user tolerance study suggested that the iGesture system would provide very good recognition rates for the purposes of this experiment, and as this was an exploratory study, there was a benefit to using a working gesture system in group scenarios, which could reveal additional issues that a WoZ approach may not (see 4).

6.5.1 Related Work

Research on group interactions with shared devices present many techniques for enabling multiple users to have personalised input ability using single display group-ware (SDG). While there are many strategies for enabling individual users to control relevant parts of a display , using personal mobile devices and individual pens for example as input controllers for digital white-boards (Rekimoto, 1997), allocating specific areas on a display for individual interactions (Tse et al., 2004), or collaborative gestures where all user input is recognised using touch sensitive table-top displays (Wu & Balakrishnan, 2003; Morris et al., 2006). In addition, non-speech audio feedback is considered a valuable technique for supporting awareness of the system to users (Hancock et al., 2005). But while a large portion of the research on CSCW considers interactions with visual displays, we wanted to continue with our previous scenarios, and explore some of these strategies for multi-user interactions for secondary task interactions with audio interfaces and semaphoric gestures.

We considered the following topics:

- Types of disruptions caused when using gestures in groups
- Recognition and signalling conflicts between multiple users
- Strategies for overcoming conflicts
- Strategies for using notifications to identify individuals within groups
- techniques for enabling autonomous gestures for individual users

The experiment considers the design as the following categories of the framework and their respective parameters. Multiple values of a single parameter indicate that this will be varied in the design:

- Gestures: Style:Semaphores, object:colour band. Set:1,2,3, or 4, complexity:simple, mapping:1-1, 1-many, many-1.
- Application Domain: Context:CSCW, multitasking, ubicomp. Tasks: low physical and cognitive requirements, ambient music control, Critical:low, Complexity:low.
- Enabling technology:Computer vision, System response:medium-fast, (see iGesture specs - since the lab has stable lighting conditions, we factor this in when determining what the accuracy rate should be for the particular system, in this

instance, we know we can expect a medium-high recognition rate from iGesture, which is over 93%). Interaction zone:1. Mobility:1

- System response: Audio output, visual display (ambient), audio recognition feedback on, reflexive feedback on, response feedback:implied (when the music changes)

This is an example of how we would describe a system in terms of the framework. In this way, we can manipulate individual settings of any parameter to change the interaction, providing a structured approach to understanding gesture system designs.

While this study is mainly exploratory in nature, our investigations did reveal some interesting findings about gesturing in CSCW scenarios with notification systems in a ubiquitous computing environment.

6.5.2 Experiment Scenario

The study was held in an office, using the iGesture system (see Chapter 3) to enable the gestures. Participants were seated at a rectangle table in the middle of the room, on which there were several puzzles, games and other non-computer based activities that they could work on as their primary tasks during the sessions. When an alert was signalled, participants would gesture to stop or start the music player, changing its current state.

Notification alerts. In on case, we investigated using a unique alert to signal the turn of each participant. In this case, each participant learnt their unique sounds in the training sessions. We used text-to-speech feedback to indicate the gesture that was recognised.

Gestures. We used two gestures, to stop and start the music player. We used different configurations of gestures and coloured markers based on the number of people in the group:

- Groups of 2: Each participant had their own alert, and gesture channel, and performed both the stop and start gesture.
- Groups of 3: Two participants shared one channel and one alert, with one person being responsible for the start gesture and the other for stop. The third person had a single channel, and was responsible for both gestures.
- Groups of 4: Participants were divided into two groups, each group of 2 shared the same alert, gesture channel and each was responsible for responding to the alert for their respective gesture.

FIGURE 6.11: A group of four participants working around the desk, performing gestures for the study.

These configurations supported a variety of configurations and strategies for dealing with group interactions. Figure 6.11 shows a group of four seated at the table where the experiment was conducted.

Distraction and confusion. One focus of our observations was to notice any distractions or confusion the participants experienced during the study, both in terms of the interruption level from the notifications, and from remembering and performing their gestures. We expected that there would be confusion getting used to the separate alerts, paying attention to the music to know if it was their gesture, and performing the correct gesture, however this was worked out in the early stages of the sessions. Participants rapidly became accustomed to their condition.

Tolerance for iGesture. As in our second study on tolerance for errors, a keyboard was provided away from the primary task area, so that participants had a choice to either gesture, or use the keyboard if they became frustrated with the gestures. The keyboard was placed in different locations depending on the groups: at the side of the table when there were two or three participants, and on a different table in the condition where there were four participants. In this case, since only one person had comfortable access to the keyboard, the other participants were instructed that they could interrupt the person sitting next to the keyboard to reach the keyboard when they were frustrated with the system. This happened rarely, as most people said that they were not comfortable disturbing the keyboard person, even though this was agreed on at the start of each session.

Participant comments. Participants were very positive about the interaction. Most did not use the reflexive feedback window, and all but two ever asked the designated keyboard operator to hit the keys for them. All participants said that the gestures were not distracting to their primary task, and all were able to seamlessly interact with the system, responding to their alerts and performing the gestures without difficulty.

6.5.3 Analysis

Based on our observations, and participant feedback, we found that most of our strategies were successful for dealing with some of the potential problems in group interaction scenarios like identifying individuals and their gestures. To address the identity issue, an individual person can use unique set of gestures or coloured markers (with the iGesture system) for signalling their own interactions with the system. Alternatively, one can employ specific notifications that are issued for a particular individual, which can be used to alert the system that this person will be gesturing. These strategies were quickly adapted by participants, causing only brief confusion during the first few trials, and indicate that in a group situation, gesture can still enable secondary tasks that are not disruptive. Our observations also indicate that while this was a CSCW environment, that it was also a ubiquitous environment, suggesting that a parameter to indicate group interactions could be added to the application domain category, under the sub category of interaction context. However, with any domain, there will be additional elements to be described as part of the context. Even in situations where there are multiple parameters to consider in multiple domains that we can use the framework to assist in making informed design decisions, based on a weighting of the values that are compared across different parameters. In addition, we note that a description of the modality of the feedback sub category would also be required to provide a clearer description of the interaction, as would the object parameter, which could use an additional parameter to indicate number of input objects used. In the case of this experiment, we consider the two different coloured bands as the objects used for tracking the gestures.

6.5.4 Discussion

This study presents an example of using the framework to define a system and to identify specific characteristics within novel domains that effect the interaction. In this case, we identified several strategies that could be used to solve some of the problems associated with identity in group interactions with iGesture. Results from this formative study can also be used to inform designs using other enabling technologies to recognised gesture input. For example, it may be reasonable to assume that if we use touch based input for semaphoric gestures, under similar variables within the framework, the same strategies for establishing user identity for the gestures would also apply.

6.6 Chapter Summary

In this chapter, we presented several examples of using the framework to guide research and design, and three studies, that contributed to the verification of the framework. These experiments demonstrated how new information could be used to extend the categories and parameters in the framework, and provided an example applying the framework to comparing different systems under similar criteria. The goal of the framework is to provide a perspective from which we can view different gesture interactions using the same parameters, leading to a more theoretical, and methodological approach to making design decisions. We next provide an overview of future research that we hope to pursue based on extending the work conducted and presented in this dissertation.

Chapter 7

Future Work: Enhancement and the Science of Design

"In the attitude of silence the soul finds the path in a clearer light, and what is elusive and deceptive resolves itself into crystal clearness. Our life is a long and arduous quest after Truth."

Mahatma Gandhi

7.1 Are We There Yet? Enhancement Articulated

In this section we describe how the research in this dissertation demonstrates our vision of the Enhancement stage of HCI, as introduced by Andrew Dillon (Dillon, 2002). We also propose the need for determining critical parameters for gesture interaction systems to identify and describe user goals. We next discuss the Enhancement stage of HCI.

7.1.1 Enhancement stage HCI

Andrew Dillon introduces the notion of a third stage of HCI - the Enhancement stage - as being a prescriptive discipline, a step forward from the evaluative discipline that has been the dominant approach for HCI to date. Dillon describes stage 1 as empirically based, focusing on interface design and the methodological tradition, where user testing and experimental trials were conducted to demonstrate that interface features worked, but with little attention towards investigating why they work. Stage 2, referred to as modelling interaction and the theoretical tradition, is described as being driven by attempts to generalise user performance data into theory that explains some form of human-computer interaction.

The third stage, described as beyond usability, aims to enhancement the design of augmenting technologies, but has not yet been attained. Dillon (2002) describes enhancement as:

> "HCI's ability to lead the design of technologies that truly empower users, to support them in the performance of tasks that would be impossible otherwise, or to enable users to overcome limitations in their own capabilities that hinder their development".

Through enhancement, Dillon hopes that HCI can become more predictive than evaluation, and can contribute to the identification and analysis of scenarios where new technological forms can enhance human capabilities. The notion of an enhancement stage to HCI is motivated by a term used by Douglas Englebart, inventor of the mouse, who called for technology to be used to augment human intellect (Englebart, 2002). In our research, we attempt to expand on this concept and take a preliminary step towards demonstrating 3rd stage HCI research. Through our proposed classification and framework for understanding gestures as a computer interaction technique, we hope to support researchers and practitioners in developing usable gesture interactions while avoiding the pitfalls of point designs. One approach we consider involves developing methods and techniques for evaluating and predicting how gestures can best be used to enhance interactions before they are even designed.

7.2 Recording and Accessing Information: Claims Libraries

Results from experiments related to parameters in the framework, however to extend our work knowledge about gestures and their application to notification system interactions led to the consideration of claims libraries as a repository for knowledge. From our research, we have learnt that interaction mode can be a factor that influences notification system interactions. McCrickard & Chewar (2005) describe a claims library for the storage of knowledge gained about notification systems. While the library is intended for addressing claims made about notification systems, we propose that gestures can be included in a claims library to present claims that are related to interaction modes and their effects on notification systems. We consider the following points in our proposal:

1. Can we incorporate gesture systems into a claims library as independent systems for which claims can be made and accessed by researchers and practitioners?
2. How can we link systems of interaction techniques such as gestures to interactive systems such as notification systems and address how different settings of each can effect the interactions?

3. Can we consider the use of the ePrints system as a repository for claims and if so, how would we design such a library to enable claims about individual systems and interactions between systems to be interactive?

This is still in the very early conceptual stages, however further investigation into its appropriateness for a claims library are under way.

7.3 Ubiquitous Computing

The trend in computing that we feel is most relevant for benefiting from the enhancement stage of HCI is this concept of ubiquitous computing (ubicomp). When considering ways to describe ubiquitous computing, we ask how today's vision has been altered from the original vision provided by Weiser (1993). Given that we are closer to realising this vision, at least in terms of the advancements in perceptual and pervasive technologies, it may be necessary to consider this domain from a new perspective of HCI research. We feel that this can be described as the enhancement stage.

The traditional paradigm for conducting HCI research, which was based on the desktop computing model may not be appropriate for perceptual or ubiquitous computing designs. Weiser suggested that we could use traditional methods to design such systems:

> "The research method for ubiquitous computing is standard experimental computer science: the construction of working prototypes of the necessary infrastructure in sufficient quantity to debug the viability of the systems in everyday use, using ourselves and a few colleagues as guinea pigs. This is an important step towards insuring that our infrastructure research is robust and scalable in the face of the details of the real world." (Weiser, 1993)

But while Weiser's vision was a motivational force behind developing a ubiquitous computing paradigm, we suggest that a possible reason for much of the research in ubicomp remaining prototypical, experimental and poorly defined may be because researches are still using 1st and 2nd stage HCI approaches, as Weiser stated, where we perform our tests on working prototypes.

There are however some researchers, such as Rodden et al. (2004), who consider ubiquitous computing as an evolutionary concept, one that can evolve using current computing technology over time enabling the integration of the old with the new. This approach is a practical and methodical means towards the end of achieving a working ubiquitous computing systems. However, much is left to the imagination in terms of translating this concept to working systems. In stead, much of what we see in terms of ubicomp remains a treat for researchers, and those who have the opportunity to access cutting edge technologies. In addition, if we don't want to risk building expensive prototypes that end up as point designs, we need an approach that is based on evolving the desktop paradigm into the ubiquitous scenario.

Rather than reinventing computer interactions, the evolutionary approach would enable researchers to slowly move the desktop paradigm out into the environment, increasing the distance with which people can control their computers using the pervasive and perceptual computing technologies that are currently available. In this way, we can conduct extensive research to enable us to confidently predict which features would be best suited to ubiquitous computing without having to build such systems first.

7.3.1 Perceptual Computing

Perceptual computing presents a new direction of ubiquitous computing research where human abilities and their social needs are considered integral in the design of systems (Crowley & Jolle Coutaz, 2000). One way in which we approach this study is to consider how this style of human-human style perceptual input can begin to enhance our computer interactions. In our work, we focus on the use of computer vision technology and hand gestures as a means of increasing the distance with which users can interact. However, while we have cameras that can easily be incorporated into our computing systems and programs, there are many limitations in the state of the art vision technology that make it difficult or currently impossible to recognise complicated gestures. But does this mean that we cannot experience the types of enhancements that are possible using vision enabled gestures? If we consider state of gestures in terms of their availability for everyday computing use, we envision many valuable interactions, however these are often not tested in real life situations. We see a similar situation with voice recognition, where any Windows or Mac OSx operating system user has voice control capabilities on their computers, however these are rarely used. In a study we conducted, out of 106 university computer science students, roughly 1/4 of the participants stated that they use voice commands on their computers. However these results do not suggest that this has been developed with enhancement in mind. The enhancement approach we discuss in this work considers current technology and different ways in which it can enhance our computing experiences. Through applying the methodological approach to gestures presented in the framework, we can begin to identify and address issues of technological limitations, and the appropriate application of current technology in its current state to create practical enhancements to our everyday computing experiences.

7.3.2 Gestures as Interaction Systems

When we think of perceptual interaction interfaces, which is not yet at the stage of enabling plug and play gestures, we must also consider the entire system that is involved in creating gesture interactions. Based on our classification, a gesture system would require an input device, and processing software to recognise the gestures. With this approach, we can begin to consider what some of the critical parameters are that can express the goals of this interaction. While we have identified several parameters within the structure of the framework (Chapter 5, further attention is intended to focus on determining critical parameters for gesture systems.

7.4 Enhancement in Action

We discuss our approach to the enhancement stage and how it developed over this thesis. This research began with our investigation of the iGesture system, and gestures for desktop computer interactions. During our investigations, we realised that our research would be constrained by the system, however we wanted to understand gestures from the human perspective, and not be restricted by the limitations of the system such as accuracy, and gesture sets. This led to our taking a different approach. We next describe how our work evolved through the 3 stages of Dillon's HCI research.

7.4.1 1st and 2nd Stage Approaches

Since we had a working gesture system, the 1st stage approach of HCI according to Dillon, would have required continued research with the iGesture system, with a focus on the user interface: improving the design and layout of the buttons used for the different settings, or trying to improve the general layout of the GUI. While this approach would be valuable at the stage where we were content with the other areas of the system, this would not provide us with any significant improvements that could quickly enhance the interactions beyond providing a pleasant and usable interface to a system for which we still needed to determine a functional value.

Taking the 2nd stage approach, we conducted usability studies on the working system, however, according to Dillon, "we are a long way from having sufficient theoretical power to predict many of the user issues that are important to usable systems design, but we are no longer completely dependent on user testing to determine the design alternatives we consider" (Dillon, 2002). That is, while we could have conducted additional studies with the system to inform on how we could better design iGesture in future iterations, the fact remained that even with the most cutting edge vision recognition technology, there are still errors in recognition, fluctuations due to lighting changes, and a host of other issues that are yet unresolved by vision researchers.

7.4.2 3rd Stage Approach

Having realised that there were inherent problems in computer vision that was beyond the scope of this research, we took a different approach: and considered users and their requirements for a gesture recognition system, and what they could tolerate in terms of the limitations that are inherent in the technology. Even though we had the iGesture system available to use for our studies, we decided that a more predictive approach was to use a series of Wizard of Oz studies to better learn what the fundamental issues of gesture interactions are in terms of enhancing the user experience rather than simply evaluating the status quo.

There were two main issues that we wanted to understand about the gesture interactions. First, we wanted to respond to critics regarding the lack of functional utility of semaphoric gestures, and second, while the system could achieve at least 93% accurate gesture recognition, we were not sure that this would be sufficient to provide a usable system. Both these issues were addressed using the WoZ approach, where we conducted 2 studies to attempt to answer these questions.

For the first study (see Chapter 3), we determined that coarse grained gestures would be best suited for secondary task interactions within a notification system and conducted our experiment to assess if gestures could improve user's performance and subjective ratings over a keyboard. The study was situated within a ubiquitous computing scenario in which a secondary task was to be controlled from a computer located at a distance from the user's primary task. The results of the study showed that gestures improved secondary task performance while reducing the distraction caused to the primary task during multitasking situations.

For the second study (see Chapter 4), we looked at user tolerances for error rates in gesture recognition. Based on our previous research with iGesture, we explored error rates in relation to several factors that were possible influences on user tolerance levels including the level of access users had to an alternative input mode, characteristics of the primary and secondary tasks, error types, error rates and system response speed. Results of this experiment demonstrated that error rates alone do not influences user tolerance. The study suggested that users will alter their tolerance levels according to a trade-off, where benefits of distance interactions must outweigh the cost imposed by system performance issues. Benefits were shown to be eyes-free, distance interactions, and improved task performance. However, users also prefer to use a direct-input device if it is available, suggesting that there must be specific contexts where gestures can be deployed successfully. While these represent a small fraction of the potential issues in understanding gesture interactions, they do support one interpretation of conducting enhancement stage to HCI.

7.4.3 Critical Parameters

One of the key contributions of this framework is potential introduction of measurable and manipulatable parameters, which enable researchers and practitioners to approach the study of gestures from an engineering perspective: Systems can be built, evaluated and compared using an established set of parameters as Newman (1997) suggests. Critical parameters are used in most engineering fields to ensure improvements, rather differences in successive system designs, and should be considered as performance targets, rather than evaluation parameters to be used at the outset of the design. Chewar et al. (2004) demonstrated this use of critical parameters as a means of describing notification systems using interruption, reaction and comprehension. It is through a similar approach that we propose critical parameters as the working units of measurement which were derived from this framework. This supports methodological approach to designing and understanding gestures as a human interaction technique, where gestures are viewed as systems of interaction techniques that can be described through a set of critical parameters. Future work is intended to investigate a selection of critical parameters to describe gesture interaction systems.

7.5 Using the Framework

Based on the results of this research, we can refer to several elements of a gesture based interaction system and scenarios in order to predict the success of a given system design and implementation. If we consider the iGesture system, experiment results suggest that the system provides a sufficient level of accuracy and system response to be considered for use in interaction scenarios.

While there are other elements of the gesture system that still require improvements, such as implementing more robust recognition and tracking algorithms, we can still predict that in the multitasking situation where the user would like to be able to control their music player while performing non-computer related secondary tasks, that gestures can improve secondary task performance, reduce distraction to the primary task, and create a more satisfying interaction experience for the user.

The value of prediction rather than simple evaluation encapsulates the enhancement stage of HCI. Prediction can occur if we consider what the users want before we go ahead and build it. There is enough technology available for us to estimate what performance measures are, and how ubiquitous devices can work together, however until we determine how to predict what is required in a design, we will not evolve past the 2nd stage of HCI design processes and methods.

Some of the questions that remain unanswered include what types of tasks users want to be able to perform using gestures, what interaction modes are appropriate for controlling

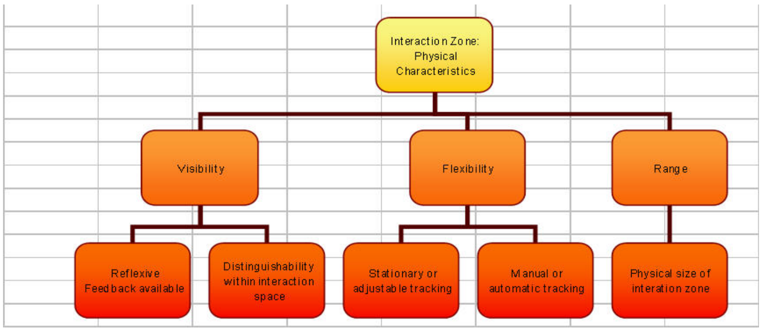

FIGURE 7.1: The diagram show some of the concepts that we would like to explore to describe an interaction zone.

those tasks, how can we deal with conflicts between different input devices when multiple input modes are available.

7.6 Defining Interaction Zones

One area that we next plan to investigate is the definition of an interaction zone. While this is a parameter within our framework, we have limited understanding of how an interaction zone can be described in terms of its size, sensitivity, and other factors that can lead towards building a more concrete method for describing the interaction zone of perceptual computing input. Figure 7.1 shows preliminary thoughts on the various characteristics that we could use to quantify this parameter.

7.7 Summary of Contributions

We propose several contribution to the field of HCI research, and several individual contributions within each chapter that further contribute knowledge and techniques that can be applied to future work. We list the key contributions here:

1. Classification of gesture research: Provides a cohesive perspective from which to understand gesture interactions. Promotes the view of gestures as a cohesive study of interaction techniques.
2. Empirical results: Aims to develop functional semaphoric gestures
3. Empirical results: Towards understanding user tolerances for gesture recognition errors, and related contexts of use.
4. The Framework: Provides a practical approach to designing gesture interaction systems, and to guiding future research

5. HCI Research: Demonstrates an approach to HCI research that supports the enhancement stage of HCI (discussed in Chapter 7).

6. Science of Design: Demonstrates an approach to the integration of qualitative and quantitative research methodologies towards the development of a science of design, discussed in Chapter 7.

7. Development of Framework: Contributes a model of viewing interactions that can be extended beyond gestures to address other interaction techniques and modes. This would require an alteration of the framework, which is discussed in Chapter 7.

7.8 In Summary

In this dissertation, we presented work that led to the development of a theoretical framework for designing and researching gesture based interactions. Our classification of the literature led to the categories proposed for understanding gestures as a human interaction technique, and was developed to enable us to gain a more theoretical perspective on the field of gesture interactions.

With the classification in place, we began investigate gesture interactions to explore the relationships between the categories, their characteristics, and their affect on interactions. Our research combines qualitative and quantitative methods, in what is referred to as qualitative positivist research methods (QPR). This approach enabled us to uncover important relationships that link the user experience to design features of a gesture system.

After conducting several studies, we determined that a framework would enable us to amalgamate information gained through our investigations, and lead us closer to our goal of understanding gestures from a more theoretical perspective. We used a grounded theory method to develop the categories, and parameters presented our theoretical framework. Using the framework to guide our research, we demonstrated how experiments can be conducted to verify the propositions proposed in the framework, and how new areas of research can be uncovered, and how using the framework to understand and compare systems can assist designers in making measurable improvements to existing system designs. Finally, we proposed several new directions that we hope can enable the continued development of the framework, and of the directions towards conducting enhancements stage HCI research and design, hopefully in the near future.

Appendix A

Literature Review and Classification

A.1 Coded Data

The following tables present the codes that were used in our analysis of the literature reviewed in this research.

	A	B	C	D	E	G	I
1	Author	Title	Year	Study	StudyDetails	Eval	Research
2	Allport	Issues of gestural navigation in abstract information spaces	1995	None/Unspecified		None	Interface/ Application
3	Alpern	Developing a car gesture interface for use as a secondary task	2003	Wizard of Oz - Usability Issues	woz	Study	Prototype
4	Amento	The sound of one hand: a wrist-mounted bio-acoustic fingertip gesture interface	2002	None/Unspecified		None	Interaction Technique/
5	Braffort	A gesture recognition architecture for sign language	1996	System Accuracy/ Performance	System-Gesture Analysis Ability	Study	Recognition Architecture
6	Barrientos	Cursive:: controlling expressive avatar gesture using pen gesture	2002	None/Unspecified		Unclear	Interaction Technique/ Method
7	Baudel	Charade: remote control of objects using free-hand gestures	1993	Usability Issues	Learnability of gestures	Study	Interface/ Application
8	Billinghurst	Put that where?	1998	.		None	Survey/Review
9	Bolt	“Put-that-there”: Voice and gesture at the graphics interface	1992	None/Unspecified		None	Interface/ Application
10	Bolt	Two-handed gesture in multi-modal natural dialog	1980	None/Unspecified		None	Prototype
11	Borchers	WorldBeat: designing a baton-based interface for an interactive music exhibit	1997	User Feedback	User feedback during development	Study	Interface/ Application
12	Bowden	Vision based Interpretation of Natural Sign Languages	2003	None/Unspecified		None	Recognition Architecture
13	Brereton	Work at hand: an exploration of gesture in the context of work and everyday life to inform the desig	2003	Exploratory	study	.	Survey/Review
14	Brewster	Multimodal 'eyes-free' interaction techniques for wearable devices	2003	Comparing Techniques	Showing advantages to using audio responses	Study	Interaction Technique/ Method
15	Buchmann	FingARTips	2004	None/Unspecified		Unclear	Interaction Technique/
16	Buxton	Issues and techniques in touch-sensitive tablet input	1985	.		None	Survey/Review
17	Buxton	CONTINUOUS HAND-GESTURE DRIVEN INPUT	1983	None/Unspecified		None	System
18	Cao	Vision Wand	2003	User Feedback	informal user feedback	User Trials	Interaction Technique/ Method
19	Chatty	Pen computing for air traffic control	1996	Usability Issues	Error Rates and acceptance	Study	Prototype
20	Cohen	QuickSet: multimodal interaction for distributed applications	1997	System Accuracy /Performance	System Recognition accuracy	Study	Prototype
21	Crowley	Perceptual user interfaces: things that see	2000	None/Unspecified		None	Survey/Review
22	Dannenberg	A gesture based user interface prototyping system	1989	.		None	Tool for gesture study or prototype
23	Davis	A perceptual user interface for recognizing head gesture acknowledgements	2001	None/ Unspecified		None	Interface/ Application
24	Eisenstein	Visual and linguistic information in gesture classification	2004	Exploratory		None	Study/Approach
25	Fails	Light widgets: interacting in every-day spaces	2002	None/Unspecified		None	System
26	Fang	Large vocabulary sign language recognition based on hierarchical decision trees	2003	System Accuracy/ Performance	System accuracy	Study	RecognitionArchitecture
27	Fisher	Virtual environment display system	1987	None/Unspecified		None	System
28	Fitzmaurice	Bricks: Laying the foundations for graspable user interfaces	1995	User Feedback	informal user feedback	User Trials	Interface/ Application
29	Forsberg	The music notepad	1998	User Feedback	informal user feedback	User Trials	System
30	Freeman	Television control by hand gestures	1994	None/Unspecified		None	Prototype
31	Freeman	Computer vision for computer games	1996	None/Unspecified		None	Recognition Architecture
32	Gandy	The Gesture Pendant	2000	None/Unspecified		None	Device
33	Goza	Telepresence control of the NASA/DARPA robonaut on a mobility platform	2004	None/Unspecified		Unclear	Interaction Technique/ Method
34	Grossman	Multi-finger gestural interaction with 3d volumetric displays	2004	None/Unspecified		Unclear	Interaction Technique/

	A	B	C	D	E	G	I
35	Gutwin	Improving interpretation of remote gestures with telepointer traces	2002	Usability Issues	Shows improvements to gesture comprehension using technique	Study	Study/Approach
36	Harrison	Squeeze me, hold me, tilt me! An exploration of manipulative user interfaces	1998	User Feedback	informal user feedback	User Trials	Prototype
37	Hauptmann	Speech and gestures for image manipulation	1989	Wizard of Oz - Usability Issues	woz	Study	Study/Approach
38	Henry	Integrating gesture and snapping into a user interface toolkit	1990	None/Unspecified		None	Tool for gesture study or prototype
39	Hinckley	Synchronous gestures for multiple persons and computers	2003	None/Unspecified		None	Interaction Technique/ Method
40	Hinckley	Two-handed virtual manipulation	1998	User Feedback	informal user feedback	User Trials	Interaction Technique/
41	Hinckley	Design and analysis of delimiters for selection-action pen gesture phrases in scriboli	2005	Exploratory	study	Study	Interaction Technique/ Method
42	Hinckley	Stitching: pen gestures that span multiple displays	2005	Usability Issues	usability tests	Study	Interaction Technique/
43	Iannizzotto	Hand Tracking for Human-Computer Interaction	2001	User Feedback	user feedback	User Trials	Device
44	Jin	GIA	2004	Usability Issues	usability tests	Study	Device
45	Karam	A Study on the use of semaphoric gestures for secondary task interactions	2004	Wizard of Oz - Usability Issues	Study to look at stuff	Study	Study/Approach
46	Keates	The use of gestures in multimodal input	1998	Exploratory	Cognitive load, gesture sets and their suitability to users with different disabilities	Study	Prototype
47	Kessler	Evlauation of the CyberGlove as a whole-hand input device	1995	System Accuracy/Performanc e	sensitivity, performance and accuracy of gloves	Study	Study/Approach
48	Kettebekov	Understanding Gestures in Multimodal Human Computer Interaction	2000	Exploratory	uncovering patterns that can be used for future work in this area	Study	Study/Approach
49	Kettebekov	Exploiting prosodic structuring of coverbal gesticulation	2004	Exploratory	investigating structure of gestures	Study	Study/Approach
50	kjeldsen	Toward the use of gesture in traditional user interfaces	1996	Usability Issues	usability within this domain	Study	System
51	Kobsa	Combining deictic gestures and natural language for referent identification	1986	None/Unspecified		None	System
52	Konrad	Gesture + play: full-body interaction for virtual environments	2003	User Feedback	informal user feedback	User Trials	Prototype
53	Koons	Iconic: speech and depictive gestures at the human-machine interface	1994	None/Unspecified		None	Prototype
54	Kopp	Towards integrated microplanning of language and iconic gesture for multimodal output	2004	None/Unspecified		None	Recognition Architecture
55	Kreuger	VIDEOPLACE—an artificial reality	1985	None/Unspecified		None	System
56	Krum	Speech and gesture multimodal control of a whole Earth 3D visualization environment	2002	System Accuracy/ Performance	testing accuracy of the device	Study	Interface/ Application
57	Kuzuoka	GestureCam: a video communication system for sympathetic remote collaboration	1994	Usability Issues	usability in different scenarios	Study	System
58	LaViola	Hands-free multi-scale navigation in virtual environments	2001	Wizard of Oz - Usability Issues	woz-exploring type of scenario most suited for this interaction	Study	Interaction Technique/ Method
59	Lee	The control of avitar motion using hand gesture	1998	System Accuracy/ Performance	system accuracy	Study	System
60	Lenman	Using marking menus to develop command sets for computer vision based hand gesture interfaces	2002	User Feedback	informal user feedback	User Trials	Prototype
61	Lenman	Computer Vision Based Recognition of Hand Gestures for Human-Computer Interaction	2002	None/Unspecified		Unclear	Prototype
62	Long	Implications for a gesture design tool	1999	System Accuracy/ Performance	rates of recognition for the system	Study	Tool for gesture study or prototype

	A	B	C	D	E	G	I
63	Lumsden	A paradigm shift: alternative interaction techniques for use with mobile & wearable devices	2003	Usability Issues	usability and gesturing accuracy of the user	Study	Interaction Technique/ Method
64	Maes	The ALIVE system: wireless, full-body interaction with autonomous agents	1997	None/Unspecified		None	System
65	Minsky	Manipulating simulated objects with real-world gestures using a force and position sensitive screen	1984	None/Unspecified		None	Prototype
66	Moyle	The design and evaluation of a flick gesture for 'back' and 'forward' in web browsers	2003	Comparing Techniques	comparison of gestures to back/foreward button	Study	Interaction Technique/ Method
67	Nickel	Pointing gesture recognition based on 3D-tracking of face, hands and head orientation	2003	Comparing Techniques	com[arison shows their 2 camera method improves accuracy	Study	System
68	Nishino	3D object modeling using spatial and pictographic gestures	1998	User Feedback	informal user feedback	User Trials	Interaction Technique/
69	Nishino	Interactive two-handed gesture interface in 3D virtual environments	1997	Comparing Techniques	comparing single to 2 handed, show benifits of 2 hand	Study	Interface/ Application
70	Osawa	Immersive graph navigation using direct manipulation and gestures	2000	None/Unspecified		None	System
71	Ou	Gestural communication over video stream: supporting multimodal interaction for remote collaborative	2003	Usability Issues	system performanc, accuracy and usability tests	Study	System
72	Oviatt	Perceptual user interfaces: multimodal interfaces that process what comes naturally	2000	.		None	Survey/Review
73	Paradiso	Sensor systems for interactive surfaces	2003	None/Unspecified		Unclear	System
74	Paradiso	Tracking contact and free gesture across large interactive surfaces	2000	None/Unspecified		None	Interaction Technique/ Method
75	Paradiso	Interfacing to the foot: apparatus and applications	2000	None/Unspecified		None	System
76	Pastel	Demonstrating information in simple gestures	2004	None/Unspecified		None	Interface/ Application
77	Pastel	A gesture-based authentication scheme for untrusted public terminals	2004	None/Unspecified		None	Device
78	Pausch	Tailor: creating custom user interfaces based on gesture	1990	None/Unspecified		Unclear	System
79	Paiva	SenToy in FantasyA: Designing an Affective Sympathetic Interface to a Computer Game	2002	Wizard of Oz - Usability Issues	woz: to determine form factor for toy and emotions that are	Study	Interface/ Application
80	Pavlovic	Gestural interface to a visual computing environment for molecular biologists	1997	None/Unspecified		None	Interface/ Application
81	Pavlovic	Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review	1996	.		None	Survey/Review
82	Pickering	Gesture recognition driver controls	2005	.		None	Survey/Review
83	Pierce	Comparing voodoo dolls and HOMER: exploring the importance of feedback in virtual environments	2002	Comparing Techniques	comparison of glove to joystick for interactions	Study	Study/Approach
84	Pirhonen	Gestural and audio metaphors as a means of control for mobile devices	2002	Comparing Techniques	comparing gesture/audio with regular pda use	Study	Study/Approach
85	Quek	Toward a vision-based hand gesture interface	1994	None/Unspecified		Unclear	Study/Approach
86	Quek	Multimodal human discourse: gesture and speech	2002	Exploratory	analyzing gesticulations	Study	Study/Approach
87	Reilly	Applications of face and gesture recognition for human-computer interaction	1998	Usability Issues	usability studies for fatigue and general usablity with impaired users	Study	Interface/ Application
88	Rekimoto	Pick-and-drop: a direct manipulation technique for multiple computer environments	2003	None/Unspecified		None	Interaction Technique/
89	Rekimoto	Smart Skin	1997	None/Unspecified		None	Prototype
90	Rekimoto	PreSense: interaction techniques for finger sensing input devices	2002	None/Unspecified		None	Interaction Technique/ Method

	A	B	C	D	E	G	I
91	Rhyne	Dialogue management for gestural interfaces	1987	None/Unspecified		None	Interface/ Application
92	Robbe	An empirical study of speech and gesture	1998	Usability Issues		Study	Study/Approach
93	Roy	Gestural human-machine interaction for people with severe speech and motor impairment due to cerebra	1994	Exploratory	exploratory research - elicitation of emotions and gestures from impaired users	Study	Study/Approach
94	Rubine	Specifying gestures by example	1991	System Accuracy/ Performance	Recognition rates of the system	Study	Tool for gesture study or prototype
95	Rubine	Combining gestures and direct manipulation	1992	None/Unspecified		None	Interaction Technique/
96	Sagawa	Description and recognition methods for sign language based on gesture components	1997	System Accuracy/Performanc e	accuracy of recognition	Study	Recognition Architecture
97	Schapira	Experimental evaluation of vision and speech based multimodal interfaces	2001	Usability Issues	evaluating on comparing selection strategies for accuracy and usablity	Study	Study/Approach
98	Schiphorst	Using a gestural interface toolkit for tactile input to a dynamic virtual space	2002	None/Unspecified		None	Tool for gesture study or prototype
99	Segan	Gesture VR: vision-based 3D hand interace for spatial interaction	1998	None/Unspecified		None	System
100	Segan	Video acquired gesture interfaces for the handicapped	1998	None/Unspecified		None	System
101	Silva	Evaluation of a visual interface for gesticulation recognition	2003	Usability Issues		Study	Interface/ Application
102	Sinclair	Salt and Pepper	2002	None/Unspecified		Unclear	Interaction Technique/ Method
103	Smith	The radial scroll tool: scrolling support for stylus- or touch-based document navigation	2004	Comparing Techniques	comparing scrolling methods for speed etc	Study	Prototype
104	Song	Developing an efficient technique of selection and manipulation in immersive V.E.	2000	Usability Issues	task time and usability	Study	Interaction Technique/
105	Stotts	FaceSpace	2004	None/Unspecified		Unclear	Interface/ Application
106	Streitz	i-LAND: an interactive landscape for creativity and innovation	1999	System Accuracy/ Performance	on the system not the gestures	Study	Prototype
107	Sturman	A design method for “whole-hand” human-computer interaction	1989	.		None	Survey/Review
108	Sturman	A Survey of Glove-based Input	1994	.		None	Survey/Review
109	Sturman	Hands-on interaction with virtual environments	1993	None/Unspecified		None	Interaction Technique/ Method
110	Swindells	That one there! Pointing to establish device identity	2002	None/Unspecified		None	System
111	Sutherland	Sketchpad	1963	None/Unspecified		Unclear	System
112	Thalmann	The virtual human as a multimodal interface	2000	None/Unspecified		None	Interface/ Application
113	von Hardenberg	Bare-Hand Human-Computer Interaction	2001	Comparing Techniques	comparing techniques	Study	Interaction Technique/Method
114	Ward	Dasher—a data entry interface using continuous gestures and language models	2000	None/Unspecified		Unclear	Interface/ Application
115	Weimer	A synthetic visual environment with hand gesturing and voice input	1989	None/Unspecified		None	Interface/ Application
116	Wellner	The DigitalDesk calculator: tangible manipulation on a desk top display	1991	None/Unspecified		None	Prototype
117	Westeyn	Georgia tech gesture toolkit: supporting experiments in gesture recognition	2003	User Feedback	informal user feedback	User Trials	Tool for gesture study or prototype
118	Wexelblat	An approach to natural gesture in virtual environmeents	1995	Exploratory		Study	Prototype
119	Wexelblat	Research challenges in gesture: open issues and unsolved problems	1997	.		None	Survey/Review
120	Wilson	Between u and i: XWand: UI for intelligent spaces	2003	System Accuracy/ Performance		Study	Interface/ Application

	A	B	C	D	E	G	I
121	Wolf	Gesturing with shared drawing tools	1993	Exploratory	looking at how people use gestures in cscw	Study	Tool for gesture study or prototype
122	Wu	Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays	2003	User Feedback	informal user trials	User Trials	Interaction Technique/ Method
123	Zeleznik	UniCam—2D gestural camera controls for 3D environments	1999	User Feedback	development based on user input	Study	Interaction Technique/ Method
124	Zhao	Simple vs. compound mark hierarchical marking menus	2004	Comparing Techniques	Comparing to compound marks for speed, accuracy	Study	Interaction Technique/ Method
125	Zimmerman	A hand gesture interface device	1995	None/Unspecified		None	Device
126	Zimmerman	Applying electric field sensing to human-computer interfaces	1986	System Accuracy/ Performance	on system performance and accuracy	Study	Interface/ Application

	K	M	O	Q	U	W	Y	AA
1	Gesture	Body Part	Conference	Domain	Input	Output	Problem	Motivation
2	Manipulative	Hand	Human Factors HCI	VR/AR	Remote Sensing/ Tracking	Desktop/General Screen	Exploring Gestures	Exploring Gestures
3	Semaphoric	Hand	Human Factors HCI	Mobile/ Pervaisve	Camera/Vision	HMD/ Stereoscopic	Specific Problems Addressed	Research Specific Issues
4	Semaphoric	Hand+Fingers	Human Factors HCI	Mobile/ Pervaisve	Sensors-On Obdy	Multiple Outputs	Novel+Improved	New and Improved
5	Sign Language	MultipleHands	Adaptive and Assitive	Communication Interfaces	Gloves	CPU	Specific Problems Addressed	Research Specific Issues
6	Semaphoric	Objects	Collaborative Virtual Environments	VR/AR	Pens/Stylus	Desktop/General Screen	NaturalInteractions ForSpecificProblesm	Other
7	Semaphoric	MultipleHands	Communications of the ACM	Desktop	Gloves	Large Screen Displays (non-projected)	Natural+Easier	Natural and Simpler
8	Manipulative	Other	Graphics + Interactions	Gesture as Interaction Mode	Camera/Vision	CPU	Exploring Gestures	Exploring Gestures
9	Deictic	MultipleHands	UIST + Interfaces	3D Graphics	Gloves	Desktop/General Screen	Natural Interactions	Natural Interactions
10	Manipulative	Hand	Graphics + Interactions	Ubiquitous/ Smart Rooms/ Appliances	Sensors-On Obdy	Large Screen Displays (non-projected)	New+NaturalInteractionsForSpecificProblems	Other
11	Multiple Styles	Objects	Human Factors HCI	Ubiquitous/ Smart Rooms/ Appliances	Remote Sensing/ Tracking	Multiple Outputs	Novel+Specific Problems	Other
12	Sign Language	Hand	Computer Vision	Communication Interfaces	Camera/ Vision	Desktop/General Screen	Exploring Gestures	Exploring Gestures
13	Multiple Styles	Other	UIST + Interfaces	Gesture as Interaction Mode	Multiple Inputs	Multiple Outputs	Exploring Gestures	.
14	Semaphoric	Head+Fingers	Human Factors HCI	Mobile/ Pervaisve	Touch Surface	Audio	Specific Problems Addressed	Research Specific Issues
15	Multiple Styles	Fingers	Graphics + Interactions	VR/AR	Camera/ Vision	HMD/ Stereoscopic	ImproveExisting+ MoreNatural	Other
16	Semaphoric	Fingers	Graphics + Interactions	Desktop	Touch Surface	Portable Devices	Novel+Improved	New and Improved
17	Multiple Styles	Objects	Graphics + Interactions	3D Graphics	Pens/Stylus	Portable Devices	Novel+Improved	New and Improved
18	Multiple Styles	Hand+Fingers	UIST + Interfaces	Multiple Domains	Camera/Vision	Large Screen Displays (non-projected)	NovelInteractions	Novel Interactions
19	Semaphoric	Objects	Human Factors HCI	Desktop	Touch Surface	Desktop/General Screen	NewImprovedGestureTechniques	Other
20	Semaphoric	Objects	Multimedia	Desktop	Touch Surface	Desktop/General Screen	Novel+Improved	New and Improved
21	Multiple Styles	Other	Communications of the ACM	Ubiquitous/ Smart Rooms/ Appliances	Camera/Vision	Large Screen Displays (non-projected)	Novel+Improved	New and Improved
22	Multiple Styles	Hand+Objects	UIST + Interfaces	Desktop	Touch Surface	Desktop/General Screen	Exploring Gestures	Exploring Gestures
23	Semaphoric	Hand+Head	PUI	Desktop	Camera/Vision	Desktop/General Screen	Exploring Gestures	Exploring Gestures
24	Gesticulation	MultipleHands	Multimodal	Communication Interfaces	Camera/Vision	Multiple Outputs	Exploring Gestures	.
25	Semaphoric	Hand	Pattern recognition and Intelligence	Ubiquitous/ Smart Rooms/ Appliances	Camera/Vision	CPU	Natural+Easier	Natural and Simpler
26	Sign Language	MultipleHands	Multimedia	Communication Interfaces	Multiple Inputs	Desktop/General Screen	Exploring Gestures	Exploring Gestures
27	Manipulative	Hand	Graphics + Interactions	VR/AR	Gloves	HMD/ Stereoscopic	Natural Interactions	Natural Interactions
28	Manipulative	Objects	Human Factors HCI	Desktop	Touch Surface	Desktop/General Screen	NovelInteractions	Novel Interactions
29	Semaphoric	Objects	UIST + Interfaces	Desktop	Touch Surface	Portable Devices	Natural Interactions	Natural Interactions
30	Manipulative	Hand	Gesture + Face Recognition	Ubiquitous/ Smart Rooms/ Appliances	Camera/Vision	CPU	Natural+Easier	Natural and Simpler
31	Multiple Styles	Hand	Gesture + Face Recognition	Games	Camera/Vision	Desktop/General Screen	Novel+Improved	New and Improved
32	Semaphoric	Hand	Wearable	Multiple Domains	Camera/Vision	CPU	Novel+Improved	New and Improved
33	Manipulative	MultipleHands	Human Factors HCI	VR/AR	Gloves	CPU	NaturalInteractions ForSpecificProblesm	Other
34	Multiple Styles	Hand	UIST + Interfaces	3D Graphics	Camera/Vision	Volumetric/Other	Exploring Gestures	Exploring Gestures

	K	M	O	Q	U	W	Y	AA
35	Multiple Styles	Objects	CSCW	CSCW	Pens/Stylus	Desktop/General Screen	Improve+SpecificProblems	Other
36	Multiple Styles	Other	Human Factors HCI	Desktop	Multiple Inputs	Portable Devices	NewInteractionsWithGestures+Natural	Other
37	Manipulative	MultipleHands	Human Factors HCI	3D Graphics	Camera/Vision	Desktop/General Screen	Exploring Gestures	Exploring Gestures
38	Multiple Styles	Objects	UIST + Interfaces	Gesture as Interaction Mode	Mouse/Joystick	Desktop/General Screen	Natural Interactions	Natural Interactions
39	Manipulative	Objects	UIST + Interfaces	Ubiquitous/ Smart Rooms/ Appliances	Objects	Desktop/General Screen	NovelInteractions	Novel Interactions
40	Multiple Styles	Objects	TOCHI	VR/AR	Objects	Desktop/General Screen	Novel+Specific Problems	Other
41	Semaphoric	Objects	Human Factors HCI	Desktop	Pens/Stylus	Desktop/General Screen	Specific Problems Addressed	Research Specific Issues
42	Manipulative	Objects	UIST + Interfaces	Mobile/ Pervaisve	Pens/Stylus	Mobile Devices	NovelInteractions	Novel Interactions
43	Multiple Styles	Fingers	Communications of the ACM	Desktop	Camera/Vision	Desktop/General Screen	NovelInteractions	Novel Interactions
44	Semaphoric	Hand	Personal and ubiquitous computing	Mobile/Pervaisve	Touch Surface	Portable Devices	Novel+Natural	Other
45	Semaphoric	Hand	Human Factors HCI	Ubiquitous/ Smart Rooms/ Appliances	Camera/Vision	Multiple Outputs	Exploring Gestures	Exploring Gestures
46	Semaphoric	Hand+Head	Adaptive and Assitive	Adaptive Technology	Multiple Inputs	CPU	Novel+Improved	New and Improved
47	Multiple Styles	Hand	TOCHI	Gesture as Interaction Mode	Gloves	CPU	Exploring Gestures	Exploring Gestures
48	Gesticulation	Hand	Web Resources - General	Communication Interfaces	Camera/Vision	CPU	Exploring Gestures	Exploring Gestures
49	Gesticulation	Hand	Multimedia	Communication Interfaces	Camera/Vision	CPU	Novel+Specific Problems	Other
50	Multiple Styles	Hand	Gesture + Face Recognition	Desktop	Camera/Vision	Desktop/General Screen	Exploring Gestures	Exploring Gestures
51	Deictic	Objects	Computational Linguistics	Desktop	Mouse/Joystick	Desktop/General Screen	Specific Problems Addressed	Research Specific Issues
52	Semaphoric	Body	Human Factors HCI	Games	Camera/Vision	Desktop/General Screen	Specific Problems Addressed	Research Specific Issues
53	Gesticulation	MultipleHands	Human Factors HCI	3D Graphics	Gloves	Large Screen Displays (non-projected)	Natural+Easier	Natural and Simpler
54	Gesticulation	Hand	Multimedia	Communication Interfaces	Camera/Vision	Desktop/General Screen	Specific Problems Addressed	Research Specific Issues
55	Deictic	Hand	Human Factors HCI	VR/AR	Camera/Vision	Large Screen Displays (non-projected)	Novel+Specific Problems	Other
56	Multiple Styles	Hand	Data visualization	3D Graphics	Camera/Vision	Desktop/General Screen	Novel+Specific Problems	Other
57	Deictic	Hand	CSCW	CSCW	Camera/Vision	HMD/ Stereoscopic	Specific Problems Addressed	Research Specific Issues
58	Multiple Styles	Body	Interactive 3D Graphics	VR/AR	Multiple Inputs	HMD/ Stereoscopic	Novel+Specific Problems	Other
59	Semaphoric	MultipleHands	Virtual Reality	VR/AR	Gloves	HMD/ Stereoscopic	NaturalInteractionsForSpecificProblesm	Other
60	Semaphoric	Hand	Web Resources - General	Ubiquitous/ Smart Rooms/ Appliances	Camera/Vision	Desktop/ General Screen	Solving Specific Gesture Problems	Other
61	Multiple Styles	Hand	Human Factors HCI	Ubiquitous/ Smart Rooms/ Appliances	Camera/Vision	Desktop/General Screen	Solving Specific Gesture Problems	Other
62	Multiple Styles	Fingers	Human Factors HCI	Gesture as Interaction Mode	Touch Surface	Desktop/General Screen	Exploring Gestures	Exploring Gestures

	K	M	O	Q	U	W	Y	AA
63	Multiple Styles	Hand+Head	Web Resources - General	Mobile/ Pervaisve	Multiple Inputs	Audio	Exploring Gestures	Exploring Gestures
64	Multiple Styles	Body	Multimedia	VR/AR	Camera/Vision	Large Screen Displays (non-projected)	NaturalInteractions ForSpecificProblem	Other
65	Multiple Styles	Fingers	Graphics + Interactions	Desktop	Touch Surface	Desktop/General Screen	Specific Problems Addressed	Research Specific Issues
66	Semaphoric	Objects	UIST + Interfaces	Desktop	Mouse/Joystick	Desktop/General Screen	Easer Interactions	Simpler Interactions
67	Deictic	Hand+Head	Multimedia	Ubiquitous/ Smart Rooms/ Appliances	Multiple Inputs	CPU	Specific Problems Addressed	Research Specific Issues
68	Multiple Styles	MultipleHands	Virtual Reality	VR/AR	Gloves	HMD/ Stereoscopic	Natural Interactions	Natural Interactions
69	Multiple Styles	MultipleHands	Virtual Reality	VR/AR	Gloves	HMD/ Stereoscopic	Natural Interactions/ Specific Problesm	Other
70	Multiple Styles	Fingers	Virtual Reality	VR/AR	Sensors-On Obdy	HMD/ Stereoscopic	Novel Interactions	Novel Interactions
71	Multiple Styles	Objects	Multimedia	CSCW	Pens/Stylus	Portable Devices	Novel+Natural	Other
72	Multiple Styles	Other	Communications of the ACM	Gesture as Interaction Mode	Camera/Vision	CPU	Exploring Gestures	Exploring Gestures
73	Manipulative	Hand	Communications of the ACM	Ubiquitous/ Smart Rooms/ Appliances	Remote Sensing/ Tracking	Large Screen Displays (non-projected)	Novel Interactions	Novel Interactions
74	Multiple Styles	Body	Web Resources - General	Ubiquitous/ Smart Rooms/ Appliances	Remote Sensing/Tracking	CPU	Solving Specific Gesture Problems	Other
75	Semaphoric	Foot	Human Factors HCI	Ubiquitous/ Smart Rooms/ Appliances	Sensors-On Obdy	Audio	Novel Interactions	.
76	Multiple Styles	Fingers	Pattern recognition and Intelligence	Mobile/ Pervaisve	Touch Surface	Desktop/General Screen	Specific Problems Addressed	Research Specific Issues
77	Manipulative	Objects	UIST + Interfaces	Ubiquitous/ Smart Rooms/ Appliances	Objects	CPU	Specific Problems Addressed	Research Specific Issues
78	Semaphoric	MultipleHands	UIST + Interfaces	Adaptive Technology	Multiple Inputs	CPU	Novel+Improved	New and Improved
79	Manipulative	Objects	Personal and ubiquitous computing	Games	Objects	Desktop/General Screen	Specific Problems Addressed	Research Specific Issues
80	Multiple Styles	Other	Pattern recognition and Intelligence	Gesture as Interaction Mode	Camera/Vision	CPU	Exploring Gestures	Exploring Gestures
81	Manipulative	Hand	Pattern recognition and Intelligence	VR/AR	Camera/Vision	Projected Displays	Novel+Improved	New and Improved
82	Semaphoric	Hand	Human Factors HCI	Mobile/ Pervaisve	Camera/Vision	Projected Displays	Exploring Gestures	Exploring Gestures
83	Manipulative	MultipleHands	Human Factors HCI	VR/AR	Gloves	HMD/ Stereoscopic	Specific Problems Addressed	Research Specific Issues
84	Multiple Styles	Fingers	Human Factors HCI	Mobile/ Pervaisve	Touch Surface	Audio	Specific Problems Addressed	Research Specific Issues
85	Gesticulation	Hand	Virtual Reality	Communication Interfaces	Camera/Vision	CPU	Natural Interactions	Natural Interactions
86	Gesticulation	Hand	TOCHI	Communication Interfaces	Camera/Vision	Desktop/General Screen	Novel+Natural	Other
87	Deictic	Body	Multimedia	Adaptive Technology	Remote Sensing/ Tracking	CPU	Novel+Improved	New and Improved
88	Multiple Styles	Fingers	UIST + Interfaces	Mobile/ Pervaisve	Touch Surface	Desktop/General Screen	Novel Interactions	Novel Interactions
89	Multiple Styles	Objects	UIST + Interfaces	Ubiquitous/ Smart Rooms/ Appliances	Pens/Stylus	Desktop/General Screen	Novel+Improved	New and Improved
90	Multiple Styles	Hand	Human Factors HCI	Ubiquitous/ Smart Rooms/ Appliances	Touch Surface	Projected Displays	Novel+Specific Problems	Other

	K	M	O	Q	U	W	Y	AA
91	Multiple Styles	Objects	Interactive 3D Graphics	Gesture as Interaction Mode	Pens/Stylus	Portable Devices	Natural Interactions	Natural Interactions
92	Deictic	Fingers	Human Factors HCI	Desktop	Camera/Vision	Multiple Outputs	Exploring Gestures	Exploring Gestures
93	Semaphoric	Hand	Human Factors HCI	Adaptive Technology	Touch Surface	Multiple Outputs	Exploring Gestures	Exploring Gestures
94	Multiple Styles	Hand+Objects	Graphics + Interactions	Gesture as Interaction Mode	Touch Surface	Desktop/General Screen	Exploring Gestures	Exploring Gestures
95	Multiple Styles	Objects+Fingers	Human Factors HCI	3D Graphics	Touch Surface	Desktop/General Screen	Novel+Improved	New and Improved
96	Sign Language	MultipleHands	Pattern recognition and Intelligence	Communication Interfaces	Gloves	CPU	New Improved Gesture Techniques	Other
97	Deictic	Hand	PUI	Desktop	Camera/Vision	Large Screen Displays (non-projected)	Exploring Gestures	Exploring Gestures
98	Multiple Styles	Hand	Human Factors HCI	VR/AR	Multiple Inputs	CPU	Natural Interactions	Natural Interactions
99	Multiple Styles	Hand	Multimedia	Adaptive Technology	Camera/Vision	CPU	Novel+Improved	New and Improved
100	Multiple Styles	Hand	Multimedia	3D Graphics	Camera/Vision	CPU	Novel+Improved	New and Improved
101	Semaphoric	Hand	Human Factors HCI	Gesture as Interaction Mode	Camera/Vision	CPU	Exploring Gestures	.
102	Semaphoric	Objects	Virtual Reality	VR/AR	Camera/Vision	Desktop/General Screen	Novel+Improved	New and Improved
103	Multiple Styles	Objects	UIST + Interfaces	Desktop	Pens/Stylus	Desktop/General Screen	Improve+SpecificProblems	Other
104	Multiple Styles	MultipleHands	Virtual Reality	VR/AR	Gloves	HMD/ Stereoscopic	Easer Interactions	Simpler Interactions
105	Multiple Styles	Fingers	Hypertext	Multiple Domains	Camera/Vision	Desktop/General Screen	Specific Problems Addressed	Research Specific Issues
106	Multiple Styles	Other	Human Factors HCI	Ubiquitous/ Smart Rooms/ Appliances	Multiple Inputs	Large Screen Displays (non-projected)	Novel+Improved	New and Improved
107	Multiple Styles	MultipleHands	UIST + Interfaces	VR/AR	Gloves	Desktop/General Screen	Exploring Gestures	Exploring Gestures
108	Multiple Styles	MultipleHands	Graphics + Interactions	Gesture as Interaction Mode	Gloves	CPU	Exploring Gestures	Exploring Gestures
109	Multiple Styles	MultipleHands	ACM Information Systems Transactions	Gesture as Interaction Mode	Gloves	CPU	Exploring Gestures	Exploring Gestures
110	Deictic	Objects	UIST + Interfaces	Gesture as Interaction Mode	Objects	CPU	Easer Interactions	Simpler Interactions
111	Multiple Styles	Objects	Graphics + Interactions	Desktop	Pens/Stylus	Desktop/General Screen	NovelInteractions	Novel Interactions
112	Semaphoric	Hand	Data visualization	VR/AR	Multiple Inputs	Desktop/General Screen	Exploring Gestures	Exploring Gestures
113	Multiple Styles	Hand	PUI	Multiple Domains	Camera/Vision	Large Screen Displays (non-projected)	Natural Interactions	Natural Interactions
114	Deictic	Objects	UIST + Interfaces	Adaptive Technology	Mouse/Joystick	Desktop/General Screen	New+Improved+ Easier Interactions	Other
115	Multiple Styles	Hand	Human Factors HCI	3D Graphics	Gloves	Desktop/General Screen	Improve+SpecificProblems	Other
116	Multiple Styles	Fingers	UIST + Interfaces	Desktop	Camera/Vision	Projected Displays	Novel Interactions	Novel Interactions
117	Multiple Styles	Other	Multimedia	Gesture as Interaction Mode	Multiple Inputs	CPU	Exploring Gestures	Exploring Gestures
118	Gesticulation	Body	TOCHI	VR/AR	Multiple Inputs	Large Screen Displays (non-projected)	Exploring Gestures	Exploring Gestures
119	Multiple Styles	MultipleHands	Gesture + Face Recognition	Gesture as Interaction Mode	Camera/Vision	Multiple Outputs	Exploring Gestures	Exploring Gestures
120	Semaphoric	Objects	Human Factors HCI	Ubiquitous/ Smart Rooms/ Appliances	Multiple Inputs	CPU	SolvingSpecificGestureProblems	Other

	K	M	O	Q	U	W	Y	AA
121	Multiple Styles	Objects	Human Factors HCI	CSCW	Pens/Stylus	Desktop/General Screen	Specific Problems Addressed	Research Specific Issues
122	Multiple Styles	Fingers	UIST + Interfaces	Desktop	Touch Surface	Projected Displays	Novel+Specific Problems	Other
123	Multiple Styles	Objects	Interactive 3D Graphics	3D Graphics	Pens/Stylus	Desktop/General Screen	Novel+Improved	New and Improved
124	Multiple Styles	Objects	UIST + Interfaces	Desktop	Multiple Inputs	Desktop/General Screen	New, Improved, Easier Interactions	Other
125	Manipulative	Fingers	Human Factors HCI	Desktop	Remote Sensing/ Tracking	CPU	Novel Interactions	Novel Interactions
126	Semaphoric	Hand	Graphics + Interactions	VR/AR	Gloves	Desktop/General Screen	Natural Interactions	Natural Interactions

Appendix B

iGesture and Semaphoric Gestures

B.1 Implementing iGesture

The iGesture software conceptually consists of two subsystems. The first is responsible for the near real-time video processing and state estimation. The second is responsible for recognising semaphoric or signal based gestures. In addition, the software allows information from the video processing stage to be extracted and used for investigating non-semaphoric continuous gesture based interactions. Primarily, the iGesture software performs a number of computer vision tasks in order to extract information from the visual scene to determine if gestures are being performed. The platform has been deliberately designed to work with the Mac OSx platform and single web cam. iGesture supports a number of loosely constrained single and two-handed gestures, such as waving. These gestures can be combined into a large command vocabulary. Significantly, these gestures can be readily detected at dynamically variable distances from the camera (Hare et al., 2005).

In order to enable near real-time processing and ensure robustness, the vision subsystem is kept relatively simple. The original design of the specifications called for a system that was able to recognise coarse scale semaphoric gestures performed with one or two hands. In order to simplify the problem domain in terms of the computer vision, it was decided that coloured markers would be used rather than attempting to track raw hand motion. The system performs processing on two *channels* in parallel. These *channels* were originally designed to correspond to the left and right hand of the user. Figure B.1 illustrates the computation performed by the subsystem.

Within each channel, the input video is smoothed using a Gaussian filter and converted to a Hue, Saturation, and Value colour-space in order to de-couple intensity changes due to

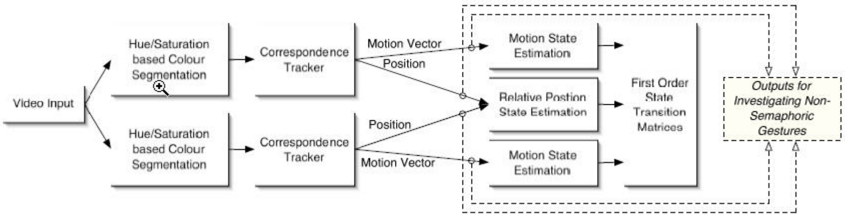

FIGURE B.1: iGesture Video Processing Subsystem

variations in lighting conditions. The colour marker corresponding to the channel is then segmented from the image by selecting pixels whose value (and 8-nearest-neighbours) have a similar hue and saturation to the marker. In order to reject noisy pixels, only the largest segmented region within a pre-determined upper and lower bound is kept for further processing. The tracking process involves comparing the shape and position of the segmented object in the current frame to the position and shape in the previous frame, and checking that the variation is within reasonable bounds. If the variation in position and shape is within the bounds defined, then a motion vector is easily created by comparing the objects position in the previous frame to its position in the current frame.

The final video processing stage involves transforming the motion vectors and marker positions into a state representation. In the current embodiment, there are 5 motion states per channel, and 4 position states which are shared across the two channels. The motion states represent whether the markers motion between the previous and current frame was either, upward, downward, left, right, or stationary. The position states represent the relative position of the two markers (if both are present) and are; marker 1 above marker 2, marker 2 above marker 1, markers approximately level, and markers really close together. From this set of states it is possible to model the motion of markers over a given number of frames as a Markov chain and determine a set of state-transition matrices that represent the chain. This set of state-transition matrices can be used to describe a semaphoric gesture.

B.2 Ambient Gestures

Some of the early work that was conducted with the iGesture system was referred to as Ambient Gestures, and we documented the system in a video, available at the links below:

- Ambient Gestures: http://www.ecs.soton.ac.uk/ãmrk03r/AmbientGestures.mov
- iGesture AirGuitar: http://www.ecs.soton.ac.uk/ãmrk03r/iGesture.mov

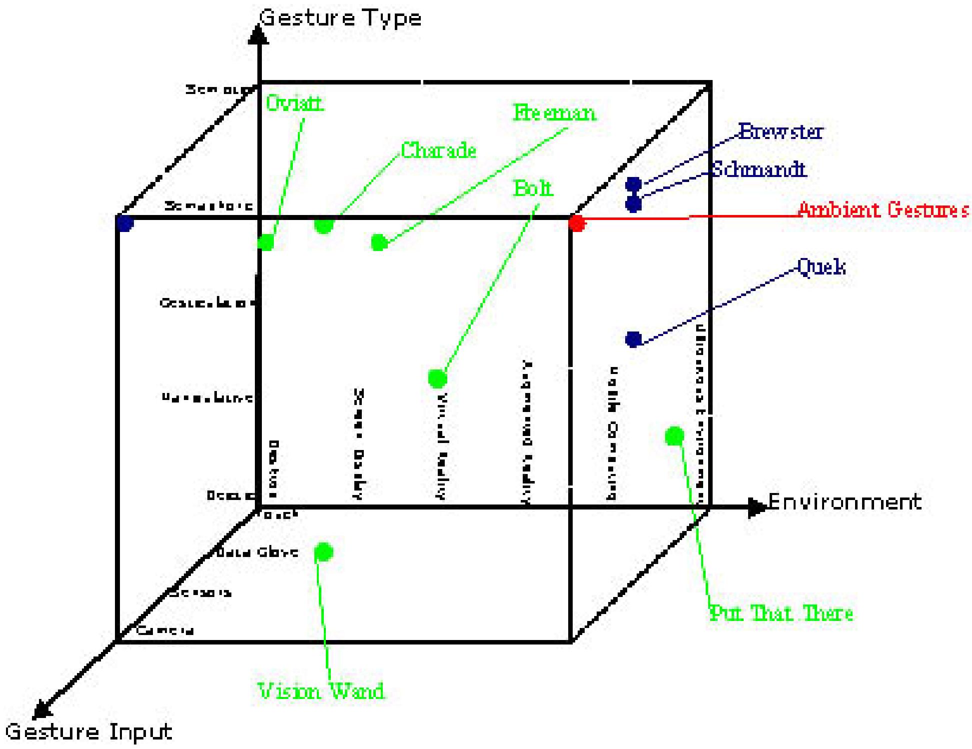

Figure B.2: The cube shows where we would place the Ambient Gestures work with respect to the early stages of our literature review and classification presented in Chapter 2.

- iGesture Web Site: http://www.ecs.soton.ac.uk/ãmrk03r/iGesture/

Appendix C

Participant Observation and Task Characteristics

C.1 The study

To try and describe secondary tasks in terms of specific characteristics within the context of multitasking scenarios, we turn our focus from controlling audio output discussed in our previous study presented in Chapter 3 to controlling visual output. In this study, we investigate tasks that are associated with controlling visual displays for secondary task interactions using gestures. We observed people in a private home for a period of 10 days. This was chosen as the time frame to enable us to observe a complete work week, as well as 2 weekends to ensure we address both work and leisure scenarios for our study, with minimum disruption to the participants normal lives. There where there were a total of 4 individuals, including the researcher who participated in the study, 2 were full time residences in the house, 1 was a frequent visitor, and the researcher who is a part-time resident of the home. Within this time, members of the household were observed conducting their normal daily activities, with discussions and interviews to attempt to uncover some of the characteristics of the tasks in multitasking situations. We conducted an empirical study that combines participant observation techniques, applying grounded theory and a mockup of a visual display to enable us to understand the types of visual information users could control as secondary task interactions with gestures. We next describe the participants, the scenarios and the methodology for our study

C.1.1 Method

A grounded theoretical approach was used in combination with the participant observation method, where the researcher resided in the house for the duration of the study, and was responsible for gathering data, conducting research activities and participating

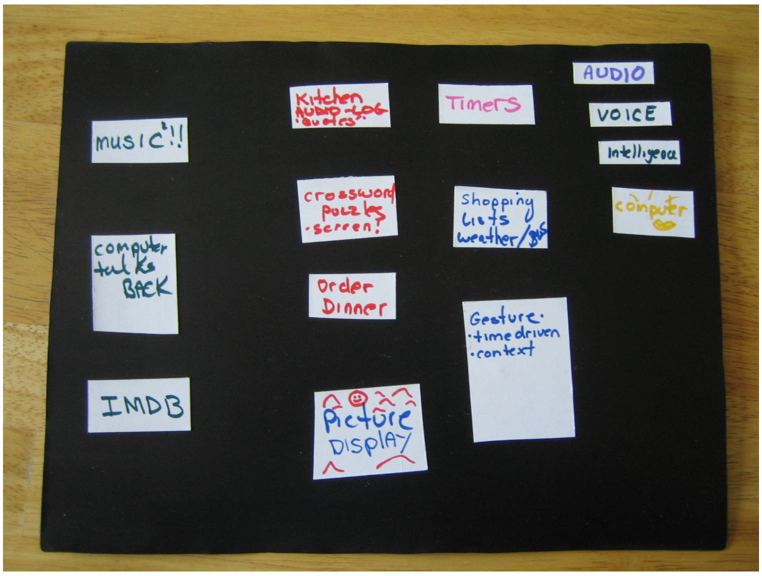

FIGURE C.1: Picture of the magnetic board used as our mockup visual display. .

in the everyday routines of the household. Field notes were coded and analysed at the end of the study, and results are presented in the Discussion Section of this chapter. However, to increase the variety of information gathered throughout the study, we devised a mockup of a visual display to record additional data from the participants on their views on visual display content that could be used for secondary task interactions.

C.1.1.1 Mockup to record Visual Display-Based Tasks

In addition to the participant observation study, we used a mockup of a visual display to further investigate secondary tasks for gesture based interactions and visual output. The mockup was of a tablet display, built out of a 8.5 x 11 inch sheet of metal, painted black on which varying sized magnetic labels could be placed. The intended purpose of this mockup was to enable the participants visualise having a display available to illicit what types of information for which they wanted to display on the tablet. With this approach, we encouraged the participants to view this as an outlet for expressing and sharing their ideas about the types of data and information that could enhance their activities at their leisure, independently of the researchers interpretation.

A set of coloured pens were located beside the display, along with an assortment of different sized blank magnetic labels on which participants could use to record anything that they felt they would like to see displayed on the screen. The screen was used in all of the common areas, and placed on a table or shelf where all participants could view it. A photograph of the mock-up that was used for the mockup is shown in Figure C.1 and the types of information that was based on the participants requests below: This activity was designed to specifically look at ambient and semi-public displays to enable us to investigate the different formats for the display that to be studied to enhance the current pool of interactions currently available for a visual system response using gestures, as with ambient or semi-public displays. The types of data included specific web sites, general sites on specific topics, and different techniques for input and output focused on the display.

C.1.2 Participants

There were 4 participants, including the researcher who took place in the study. We provide a profile of each:

1. Professional Programmer. Participant 1 is a programmer by trade, who works at home half of the time and at the office the other half. This participant uses the computer for work, primarily programming, the occasional game, research and communication. While the majority of his days are spent on the computer, he limits his off-work usage to web based research and communication tasks. This user has advanced computer knowledge and skills.

2. Student. Participant 2 is a student at a community college studying radio production and broadcasting. He has a laptop computer that he uses for school and for personal use, that uses an Ethernet connection so it remains in his room when in use. His computer skills are average, as he primarily uses the computer for communication purposes, text editing, and media applications.

3. Editor. Participant 3 does not have a home computer, but uses one only at the office during work hours. She uses the computer for document editing and communication purposes, as well as for web browsing and research and windows based games. She spends most evenings and weekends at the house.

4. Researcher and student. Participant 4 is the researcher who conducted this study is a computer science PhD student, who uses the computer for work and personal related research and activities, as well as for media access, and is considered an advanced user who is heavily dependent on the computer and the wireless Internet for both work and non-work related tasks on a laptop computer.

C.1.3 Describing Tasks

The house where the study took place has a wireless Internet connection, and no shared computing devices. The stereo is non-computer based, and there are no other devices in the home that are controlled by computer. The study focused on observing multitasking scenarios that arose during work and leisure times whenever the members were in one of the common rooms of the house, which included a lounge, recreation area and kitchen.

Primary and secondary tasks were defined as follows for our study, based on an attempt to expand on existing definitions, which are provided for dual task interactions in which both the primary and secondary tasks are situated on computer displays (e.g. Adamczyk & Bailey, 2004; McCrickard et al., 2003b; Czerwinski et al., 2004). While our description of multitasking scenarios are exploratory, we do attempt to extend the descriptions provided in computing literature to enable our investigation to consider primary tasks that are not computer based:

- A primary task requires the participants full attention, and is the central or key activity that is being conducted when an interruption occurs to indicate that a secondary task requires their attention. We describe some of the = primary tasks later in this report. These include tasks that require physical or cognitive engagement such as reading, conversing, or performing physical activities which require the users concentration.
- A secondary task is described as one that is signalled through some form of interruption at time when the user is engaged in a primary task. While a secondary task could become a primary task, we identify them as the act immediately required to respond to a notification, and one that does not require the full attention of the participant. These include physical or cognitive acts such as providing a verbal response, performing a physical action, or an attention shift for acknowledgements.

C.2 Observations

In this section, we describe the observations made throughout the study, followed by an analysis and detailed report of our results.

C.2.1 Multitasking and Work

During working hours, Participant 1 was often interrupted by the telephone, people knocking at the door, and by other tasks such as chores, (laundry, cleaning tasks, watering plants, snack and tea breaks), setting and changing music on the radio, and by lighting changes. Since this is the primary care taker of the house, he is responsible for maintenance, and care of the home. The heating is controlled by a programmable thermostat, so the temperature did not require changing. In addition, the researcher also worked from the home, noting that most of the secondary tasks that were performed while working on the computer involved turning on or off printers or other peripheral computer devices, answering the phone, checking messages or responding to interruptions from other household members. While Participant 1 spend the majority of his day working on a computer, all of the peripheral information that he wanted to access as a secondary task was conducted on his work computer, so there was little information available in terms of work and visual display based secondary tasks during work hours. Since Participant 1 and Participant 4 both worked using a computer, and were the only people who worked at home, we could not investigate non-computer based primary tasks in this study. This led to our shift in focus to investigate secondary tasks and visual outputs for gestures during leisure scenarios.

C.2.2 Multitasking and Leisure

Leisure times often included all the members of the household, who spent time in the common areas engaged in primary tasks including discussions, cooking and dining together, or socialising with their guests and working on other activities including crossword puzzles, listening to music, playing the piano or another instrument or reading as examples. They did not participate in the study when they were not in the common areas. In the leisure scenario, where participants conducted these non-computer based primary tasks, interruptions included timers signalling that food was ready, knocks at the door, telephone rings, changes in the lighting, media or stereo system changes that were signalled by a specific time or change in the state of the devices such as a c.d. finishing. In addition, participants were interrupted by unidentifiable motivations, which signalled them to suddenly stop their primary task to search for objects for no apparent reason. In these cases, the researchers would ask the participants what their motivations were, which led to our next topic, where we identify different sources for notifications in terms of internal or external interruptions.

C.2.3 Notifications

One of our observations suggests that secondary tasks for multitasking situations are typically signalled using some form of notification system: This can originate from internal or external stimuli to the participant and can elicit varying responses based on the context of the notification and on the nature of the secondary task. We describe an internal notification as implied, an idea, memory or some other form of cognitive signal that is not an explicit signal originating from the environment. We describe an external notification is an interruption that is generated through an explicit sound, visual signal or other form of explicit signal originating from a source external to the participant.

C.2.4 Responses

We observed different responses to notifications from the participants for similar types interruptions. The response to a notification appeared to be dependent on the context of both the primary and secondary task. For example, when the phone rang while participants were engaged in a primary task that demanded their full cognitive or physical attention, such as working, or dining, the interruption did not illicit a strong response from participants unless thee was some importance or expectation for the call. However, when a participant was anticipating an event, or when they appeared to be bored with their primary tasks, an interruption such as a phone call or a knock at the door would become a priority, and a primary task would be abandoned without hesitation in some cases.

C.3 Analysis

In this section, we present our analysis of the data that was gathered over the duration of the study, as well as a presentation of the types of visual display information that users recorded using the visual display mockup. We identify several characteristics of primary and secondary tasks within the multitasking situations observed in this study and present these next, followed by our intended use of this data to inform our next gesture experiment. Our approach to the analysis is based on extending the Notification system interaction model, which addresses three critical parameters that define these interactions: interruption, reaction and comprehension which we use to ground our characterisations of primary and secondary tasks for multitasking situations Chewar et al. (2004).

C.3.1 Task Criticality

We determined that the level of criticality that is associated with either a primary or secondary task is a factor that can influence the reaction that participants have to a notification. We generalise our findings into the following claim: Primary tasks that are critical will lead to a lower response to an interruption when the secondary task has a lower critical rating. Primary tasks that are not critical will lead to a response to an interruption that is reflects the level of criticality placed on the secondary task. If there are similar levels of criticality for both primary and secondary tasks, the response will be determined by factors based on the specific goals of the participant.

C.3.2 Task Relationships

In addition to task criticality, we observed that the relationships between the primary and secondary tasks also play a role in determining the level of response a participant gave to an interruption. Dependency relationship can exist between the tasks, such that a primary task may be dependent on receiving information from a secondary task before it can proceed. In this case, the secondary task takes on the critical level of the primary task, and an interruption is based on that critical level. When there is no dependency on the secondary task, then the individual critical levels can predict the response levels. A secondary task may also be dependent on a primary task, then the criticality level of the primary task may change depending on the critical level of the secondary task to enable the appropriate level of response by the participant.

C.3.3 Internal and External Notifications

Our study suggested a classification of notifications for signalling a secondary task as internal or external. An external alert is issued from the environment through perceptual or physical signals, that including ambient or visual alerts such as time and lighting changes, auditory alerts including alarms, speech or music changes and social alerts based on interruptions by people or pets for example. An internal notification involves those which originate from the self, such as remembering, hunger, and other motivations generated without any obvious external stimuli. While there is a great deal of research conducted on notification systems referenced throughout this paper, we identify internal and external as specific classes of notifications that can occur when dealing in mixed media environments as described in this study. From this observation, we next turn to a discussion on describing secondary tasks.

C.3.4 Task Complexity

Both primary and secondary tasks can be considered as having a level of complexity or difficulty associated with them associated with physical or cognitive actions (or both). We observed that when a primary task involves high physical or cognitive engagement such that the it cannot be interrupted, that the levels of response to secondary task notifications can be reduced. However, response to an interruption for a similarly complex secondary task may also be lower as participants seem to anticipate a longer interruption, and wait for an opportune time before responding and leaving their primary tasks. From this observation, we propose an measure for quantifying the level of complexity that can be associated with a secondary task to assist in making decisions about designing gestures for those tasks.

C.3.4.1 Tasks measured using interaction units

We address the level of complexity involved in completing a secondary task in terms of the number of units, or interaction actions, such as button clicks, presses, waves etc..., that is required for the interaction. We could refer to single unit of interaction such as pressing one button, or communicating one gesture to the camera as a single-response task. We could refer to a task that required two units, such as hitting two keys, or performing two individual gestures as a binary-response task ad infiniteum. In this respect, we classify secondary tasks as being simple or complex, in terms of the number of actions required to complete the task.

C.3.4.2 Gestures and secondary tasks

To apply these classification to our research into gesture based interactions, we propose a unit of measurement to enable us to apply a more methodological approach to investigating gestures. So, if we can begin to quantify the types of tasks that a gesture could be used to control in terms of descriptive complexity ratings, we can more effectively predict the value of understanding gesture as a useful interaction technique.

C.3.5 Visual Display Tasks and Gestures

In this section, we present our analysis of the types of data that participants expressed interest in having presented on the ambient display mockup to illicit responses. We discuss the types of primary tasks that were conducted, and the various classes of information or data that was suggested. In addition, we consider these as potential candidates for gesture interactions, and discuss the different display tasks in terms of the metrics discussed earlier in this chapter.

C.3.5.1 Gestures, tasks, and visual output

Group conversations was a daily occurrence during the study, and it was during these times that the mockup saw the most use. For many topics of conversation, there were instances when participants felt that having information available on the display would be useful. We discuss some of the visual based tasks that were suggested by the participants, followed by our analysis of the data in terms of their complexity ratings for use with gestures.

- Film and music web sites. For all discussions that involved film or television based discussion, any discrepancy that occurred prompted users to request that the Internet movie database (IMDB - www.imdb.com) be displayed on the screen. In addition, music based discussions often led to participants wanting information to be displayed about the specific topic or artist in question. While there were no web site that all participants would refer to for this information, a Google search was requested. In addition, there were requests for information about current events such as viewing movie listings as well as other entertainment events that were discussed within the group.

- Current affairs and history web sites. Another category of discussion that often led to the request of information was based on topics in the news, on local activities or on historic facts. For most of these queries, Google was also chosen as the main point of access for acquiring the data as was a local news site.
- Audio-visual media controller. All participants agreed that it would be useful to be able to control the stereo from the kitchen (stereo was located in the living room), since anyone who wanted to change or play music had to walk to the other side of the house to access the stereo. The mockup screen was used as an interface to the external devices, enabling distance interactions that were more complicated than simple interactions to be controlled. Telephones were also located in different parts of the house, however, participants wanted to have access to their phones from the kitchen. Often members would misplace their phone, and requested that some device be invented that could enable them to access their phone messages, or monitor who is calling from the screen.
- Miscellaneous Information and visual display data. Additional requests for information on the semi-public display included that which could assist with everyday chores and household tasks such as a timer, dictionaries and crossword puzzles on the screen, a screen to enable automatic ordering of food from local restaurants, shopping lists and weather displays. A request was also made for a picture display to be present on the screen.

C.3.6 Measuring Task Complexity

A task with a low complexity level for example would be requesting song information to be displayed on the screen. In the case where the music player is computer based, this may be achieved in a single step involving turning on the track information display for example.

A more complicated task such as searching for an historic fact using Google could require far more steps, including text input, and possibly scrolling and multiple clicks that would be too complex for using gestures to interact with a web interface.

A task of medium complexity could be accomplished in a few steps, such querying a site such as IMDB (Internet music database) to search for the name of an actor in a film for example. This would require perhaps three steps, one unit if you already had a direct link to the site on your web display, and then an additional interaction required for inputting each of the characters in your query, however, it is less complicated in terms of the actual steps than our previous example. However, this is one way in which we can assess our tasks in terms of the level of complexity they are rated for gesture use, based on the number of interaction units one has to perform. This of course can be altered through altering the interface design for example.

Participants often engaged in individual activities during leisure times, including working on various puzzles, word games, Sudoku or knitting for example, where less interactive display options were suggested, including displaying recipes, pictures, knitting patterns or other static displays to provide instructions or a reference on the screen. More ambient data displays included digital photo albums, news feeds, weather data or other media that is presented on the display. In these cases, gestures would seem to be most appropriate since the information on such displays is likely formatted for access, and some of the relevant interactions for such information would include switching between displays, searching for or requesting specific information or stopping or starting a display for example. This represents the most basic functionality of digital music players, photo albums, slide shows and television channel for example.

C.3.7 Input Modalities

Participants also suggested different ways in which they through about interacting with the display. Responses included voice and audio interactions, gestures, or standard desktop computer input controls. As the focus was primarily on secondary tasks and not input techniques, we recorded these responses to address the concerns of the participants, who wanted to know how they would be able to access and change such information if it were available.

C.4 Multitasking and Task Characteristics

We propose that secondary tasks can be classified based on several characteristics that we have identified in this study. These are discussed next.

- Secondary Task Complexity: We identify two measures of complexity when categorising a secondary task. First, we consider the number of decisions that one has to make in order to complete a single action secondary task and second, we consider the type of interaction that is required to complete the secondary task. For decision based tasks, when a single action is required, we refer to this as a single-decision task: Examples include dismissing a notification, or acknowledging that a notification has been accepted. Binary-decision tasks would require the user to indicated one of two potential responses to a notification, or tasks which provide binary options, such as selecting yes or no, ok or cancel, and play or stop for example. In the context of this research, as the number of options required to complete a single secondary increases, so does its complexity in terms of the number of gestures that are required to signal the users intent.

  The second measure of complexity involves the type of action that is required to complete the secondary task. As mentioned above, decision or selection based tasks are simple in terms of the type of response that a user must provide. However, there are simple tasks that require text based input for example, which in the context of gesture based interactions would increase the complexity of the task. Consider a binary task where the user has to signal a specific response to the computer using text entry. This would require a larger set of gestures to be learnt in order to ensure proper representation of the possible set of input options.

- Relationship to the primary task: From our investigation, we note that primary tasks and secondary tasks are either related or not. In related tasks, there exists some dependency between the tasks. For example, a primary task may depend on the completion of a secondary task before it can proceed, as would be seen in a reading task, where the page must be turned (secondary task) in order to continue to the next page. In unrelated tasks, no functional relationship exists between the primary and secondary task.

- Sensory modality requirements of the secondary task: The perceptual channel with which is required to perform a secondary task is another factor that can influence multitasking situations. Tasks that do not require the users visual attention pose potentially less demands on the users attentional resources than audio based tasks if only by fewer perceptual channels.

- Task Criticality: The criticality of a task is an important factor to consider within the context of computer interactions. While the context of an interaction can change the level of criticality of a task, a task can also be inherently critical. For example, the simple task of lowering the volume of a stereo is inherently non-critical, the need to quickly lower the volume if you are taking a call from a future employer could increase the level of criticality. Whereas inherently critical tasks are those that require precision in any situation such as ensuring you select the right button when answering the million dollar question on a game show for example.

- Task Persistence: We distinguish between tasks that are continuous or discrete. A continuous task is persistent in that the user must continually monitor an activity or display, while a discrete task is complete after the user responds to it. While it is difficult to classify something like music playing the background as a continuous task when one can simply turn it on for example and let it play, watching a clock to ensure that a specific deadline is met could be called a continuous task. To deal with this ambiguity, we claim that a continuous task is one that does not provide explicit notifications to indicate that an action must be taken, while a discrete task would imply that a notification will be issued in the event that an action must be taken.

C.5 Summary

In this participant observation study, where we incorporated grounded theory and a mockup of a visual display to record information about types of tasks could be controlled using gesture interactions, we analyse our observations based on the notification system interaction model in order to ground our research in the computing domain, suggesting several characteristics that can be used to understand a secondary task in terms of its complexity, relationship to a primary task as well as its criticality. With these classifications, we proceeded to design our next experiment, which is investigating user tolerance for gesture system recognition errors. While our observations are situated within the domain of home computing for leisure times and multitasking, we were able to provide additional characteristics of secondary task interactions that can be added to our understanding of tasks for future evaluation and investigation.

Bibliography

(2006) Eyetoy web site. Electronic Resource.

(2006) Web site: University of chicago: Mcneill lab for gesture and speech research. Electronic Resource.

Abowd, G. (1999) Classroom 2000: An experiment with the instrumentation of a living educational environment. IBM Systems Journal 38(4):508–530.

Adamczyk, P. D. & Bailey, B. P. (2004) If not now, when?: the effects of interruption at different moments within task execution. In: Proceedings of the 2004 conference on Human factors in computing systems. ACM Press, pp. 271–278.

Adams, A., Blandford, A. & Lunt, P. (2005) Social empowerment and exclusion: A case study on digital libraries. ACM Trans. Comput.-Hum. Interact. 12(2):174–200.

Allan Christian Long, J., Landay, J. A. & Rowe, L. A. (1999) Implications for a gesture design tool. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 40–47.

Allport, D., Rennison, E. & Strausfeld, L. (1995) Issues of gestural navigation in abstract information spaces. In: Conference companion on Human factors in computing systems. ACM Press, pp. 206–207.

Alpern, M. & Minardo, K. (2003) Developing a car gesture interface for use as a secondary task. In: CHI '03 extended abstracts on Human factors in computing systems. ACM Press, pp. 932–933.

Amento, B., Hill, W. & Terveen, L. (2002) The sound of one hand: a wrist-mounted bio-acoustic fingertip gesture interface. In: CHI '02 extended abstracts on Human factors in computing systems. ACM Press, pp. 724–725.

Anderson, D., Abdalla, A., Pomietto, B., Goldberg, C. & Clement, C. (2002) Senate document no. 14, distracted driving: Review of current needs, efforts, and recommended strategies. Senate Document.

Barrientos, F. A. & Canny, J. F. (2002) Cursive:: controlling expressive avatar gesture using pen gesture. In: Proceedings of the 4th international conference on Collaborative virtual environments. ACM Press, pp. 113–119.

Baudel, T. & Beaudouin-Lafon, M. (1993) Charade: remote control of objects using free-hand gestures. Commun. ACM 36(7):28–35.

Beaudouin-Lafon, M. (2004) Designing interaction, not interfaces. In: AVI '04: Proceedings of the working conference on Advanced visual interfaces. ACM Press, New York, NY, USA, pp. 15–22.

Blandford, A. & Wong, W. (2004) Situation awareness in emergency medical dispatch. International Journal of Human-Computer Studies. 61(4):421–452.

Bolt, R. A. (1980) Put that there - voice and gesture at the graphics interface. In: SIGGRAPH '80: Proceedings of the 7th annual conference on Computer graphics and interactive techniques. ACM Press, New York, NY, USA, pp. 262–270.

Bolt, R. A. & Herranz, E. (1992) Two-handed gesture in multi-modal natural dialog. In: Proceedings of the 5th annual ACM symposium on User interface software and technology. ACM Press, pp. 7–14.

Borchers, J. O. (1997) Worldbeat: designing a baton-based interface for an interactive music exhibit. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 131–138.

Boussemart, Y., Rioux, F., Rudzicz, F., Wozniewski, M. & Cooperstock, J. R. (2004) A framework for 3d visualisation and manipulation in an immersive space using an untethered bimanual gestural interface. In: VRST '04: Proceedings of the ACM symposium on Virtual reality software and technology. ACM Press, New York, NY, USA, pp. 162–165.

Bowden, R., Zisserman, A., Kadir, T. & Brady, M. (2003) Vision based interpretation of natural sign languages. In: Exhibition at ICVS03: The 3rd International Conference on Computer Vision Systems. ACM Press.

Braffort, A. (1996) A gesture recognition architecture for sign language. In: Proceedings of the second annual ACM conference on Assistive technologies. ACM Press, pp. 102–109.

Brereton, M., Bidwell, N., Donovan, J., Campbell, B. & Buur, J. (2003) Work at hand: an exploration of gesture in the context of work and everyday life to inform the design of gestural input devices. In: Proceedings of the Fourth Australian user interface conference on User interfaces 2003. Australian Computer Society, Inc., pp. 1 – 10.

Brewster, S., Lumsden, J., Bell, M., Hall, M. & Tasker, S. (2003) Multimodal 'eyes-free' interaction techniques for wearable devices. In: Proceedings of the conference on Human factors in computing systems. ACM Press, pp. 473–480.

Buchmann, V., Violich, S., Billinghurst, M. & Cockburn, A. (2004) Fingartips: gesture based direct manipulation in augmented reality. In: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and Southe East Asia. ACM Press, pp. 212–221.

Buxton, W. (1983) Lexical and pragmatic considerations of input structures. SIGGRAPH Comput. Graph. 17(1):31–37.

Buxton, W., Fiume, E., Hill, R., Lee, A. & Woo, C. (1983) Continuous hand-gesture driven input. In: Proceedings of Graphics Interface '83, 9th Conference of the Canadian Man-Computer Communications Society. pp. 191–195.

Buxton, W., Hill, R. & Rowley, P. (1985) Issues and techniques in touch-sensitive tablet input. In: Proceedings of the 12th annual conference on Computer graphics and interactive techniques. ACM Press, pp. 215–224.

Cao, X. & Balakrishnan, R. (2003) Visionwand: interaction techniques for large displays using a passive wand tracked in 3d. In: UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 173–182.

Card, S. K., Mackinlay, J. D. & Robertson, G. G. (1990) The design space of input devices. In: CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 117–124.

Card, S. K., Mackinlay, J. D. & Robertson, G. G. (1991) A morphological analysis of the design space of input devices. ACM Trans. Inf. Syst. 9(2):99–122.

Carroll, J. M. & Rosson, M. B. (1992) Getting around the task-artifact cycle: how to make claims and design by scenario. ACM Trans. Inf. Syst. 10(2):181–212.

Chatty, S. & Lecoanet, P. (1996) Pen computing for air traffic control. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 87–94.

Chen, F., Choi, E., Epps, J., Lichman, S., Ruiz, N., Shi, Y., Taib, R. & Wu, M. (2005) A study of manual gesture-based selection for the pemmi multimodal transport management interface. In: ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces. ACM Press, New York, NY, USA, pp. 274–281.

Chewar, C. M., McCrickard, D. S. & Sutcliffe, A. G. (2004) Unpacking critical parameters for interface design: evaluating notification systems with the irc framework. In: DIS '04: Proceedings of the 2004 conference on Designing interactive systems. ACM Press, New York, NY, USA, pp. 279–288.

Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L. & Clow, J. (1997) Quickset: multimodal interaction for distributed applications. In:

Proceedings of the fifth ACM international conference on Multimedia. ACM Press, pp. 31–40.

Coleman, M. (1969) In: Proceedings of the 2nd University of Illinois Conference on Computer Graphics.

Cowan, N., Elliott, E., Saults, J., Morey, C., Mattox, S., Hismjatullina, A. & Conway, A. (2005) On the capacity of attention: Its estimation and its role in working memory and cognitive aptitudes. Cognitive Psychology 51(1):42–100.

Crowley, J. L. & Jolle Coutaz, F. B. (2000) Perceptual user interfaces: things that see. Commun. ACM 43(3):54–ff.

Czerwinski, M., Cutrell, E. & Horvitz, E. (2000) Instant messaging and interruption: Influence of task type on performance. In: OZCHI 2000. pp. 356–361.

Czerwinski, M., Horvitz, E. & Wilhite, S. (2004) A diary study of task switching and interruptions. In: Proceedings of the 2004 conference on Human factors in computing systems. ACM Press, pp. 175–182.

Dannenberg, R. B. & Amon, D. (1989) A gesture based user interface prototyping system. In: Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology. ACM Press, pp. 127–132.

Davis, J. W. & Vaks, S. (2001) A perceptual user interface for recognizing head gesture acknowledgements. In: Proceedings of the 2001 workshop on Percetive user interfaces. ACM Press, pp. 1–7.

Dey, A. K. & Abowd, G. D. (1999) Towards a better understanding of context and context-awareness. Tech. rep., Georgia Institute of Technology, College of Computing.

Dillon, A. (2002) HCI and the technologies of information In HCI and the Millennium, ACM Press/Addison Wesley, chap. 20, pp. 457 – 474.

Eisenstein, J. & Davis, R. (2004) Visual and linguistic information in gesture classification. In: Proceedings of the 6th international conference on Multimodal interfaces. ACM Press, pp. 113–120.

Eliasmith, C. (2004) 'abduction' in dictionary of philosophy of mind. Electronic Source.

Elliott, G. J., Jones, E. & Barkerc, P. (2002) A grounded theory approach to modelling learnability of hypermedia authoring tools. Interacting with Computers 14(5):547–574.

Englebart (2002) Web reference for doug englebart. Electronic Resource.

Fails, J. A. & Olsen, J. D. (2002) Light widgets: interacting in every-day spaces. In: Proceedings of the 7th international conference on Intelligent user interfaces. ACM Press, pp. 63–69.

Fang, G., Gao, W. & Zhao, D. (2003) Large vocabulary sign language recognition based on hierarchical decision trees. In: Proceedings of the 5th international conference on Multimodal interfaces. ACM Press, pp. 125–131.

Fisher, S. S., McGreevy, M., Humphries, J. & Robinett, W. (1987) Virtual environment display system. In: Proceedings of the 1986 workshop on Interactive 3D graphics. ACM Press, pp. 77–87.

Fitzmaurice, G. W., Ishii, H. & Buxton, W. A. S. (1995) Bricks: laying the foundations for graspable user interfaces. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press/Addison-Wesley Publishing Co., pp. 442–449.

Forsberg, A., Dieterich, M. & Zeleznik, R. (1998) The music notepad. In: Proceedings of the 11th annual ACM symposium on User interface software and technology. ACM Press, pp. 203–210.

Freeman, W. & Weissman, C. D. (1995) Television control by hand gestures. Tech. rep., IEEE Intl. Wkshp. on Automatic Face and Gesture Recognition.

Freeman, W., Tanaka, K., Ohta, J. & Kyuma, K. (1996) Computer vision for computer games. Tech. rep., nd International Conference on Automatic Face and Gesture Recognition.

Gandy, M., Starner, T., Auxier, J. & Ashbrook, D. (2000) The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In: Proceedings of the 4th IEEE International Symposium on Wearable Computers. IEEE Computer Society, p. 87.

Ganoe, C. H., Somervell, J. P., Neale, D. C., Isenhour, P. L., Carroll, J. M., Rosson, M. B. & McCrickard, D. S. (2003) Classroom bridge: using collaborative public and desktop timelines to support activity awareness. In: UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 21–30.

Gaver, W. W., Bowers, J., Boucher, A., Gellerson, H., Pennington, S., Schmidt, A., Steed, A., Villars, N. & Walker, B. (2004) The drift table: designing for ludic engagement. In: CHI '04: CHI '04 extended abstracts on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 885–900.

Glaser, B. G. & Strauss, A. (1967) Discovery of Grounded Theory: Strategies for Qualitative Research., Aldine Transaction.

Goza, S. M., Ambrose, R. O., Diftler, M. A. & Spain, I. M. (2004) Telepresence control of the nasa/darpa robonaut on a mobility platform. In: Proceedings of the 2004 conference on Human factors in computing systems. ACM Press, pp. 623–629.

Grossman, T., Wigdor, D. & Balakrishnan, R. (2004) Multi-finger gestural interaction with 3d volumetric displays. In: Proceedings of the 17th annual ACM symposium on User interface software and technology. ACM Press, pp. 61–70.

Gutwin, C. & Penner, R. (2002) Improving interpretation of remote gestures with telepointer traces. In: Proceedings of the 2002 ACM conference on Computer supported cooperative work. ACM Press, pp. 49–57.

Hancock, M. S., Shen, C., Forlines, C. & Ryall, K. (2005) Exploring non-speech auditory feedback at an interactive multi-user tabletop. In: GI '05: Proceedings of the 2005 conference on Graphics interface. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, pp. 41–50.

Hare, J., Karam, M., Lewis, P. & schraefel, m. c. (2005) igesture: A platform for investigating multimodal, multimedia gesture-based interactions. Tech. rep., Electronics and Computer Science, University of Southampton.

Harrison, B. L., Fishkin, K. P., Gujar, A., Mochon, C. & Want, R. (1998) Squeeze me, hold me, tilt me! an exploration of manipulative user interfaces. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press/Addison-Wesley Publishing Co., pp. 17–24.

Hauptmann, A. G. (1989) Speech and gestures for graphic image manipulation. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 241–245.

Henry, T. R., Hudson, S. E. & Newell, G. L. (1990) Integrating gesture and snapping into a user interface toolkit. In: Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology. ACM Press, pp. 112–122.

Hinckley, K. (2003) Synchronous gestures for multiple persons and computers. In: Proceedings of the 16th annual ACM symposium on User interface software and technology. ACM Press, pp. 149–158.

Hinckley, K., Pausch, R., Proffitt, D. & Kassell, N. F. (1998) Two-handed virtual manipulation. ACM Trans. Comput.-Hum. Interact. 5(3):260–302.

Ho, C. & Spence, C. (2005) Assessing the effectiveness of various auditory cues in capturing a driver's visual attention. Journal of Experimental Psychology: Applied 1:157–174.

Horvitz, E. (1999) Principles of mixed-initiative user interfaces. In: CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 159–166.

Iannizzotto, G., Villari, M. & Vita, L. (2001) Hand tracking for human-computer interaction with graylevel visualglove: Turning back to the simple way. In: Workshop on Perceptive User Interfaces. ACM Digital Library. ISBN 1-58113-448-7.

Jacob, R. J. K., Sibert, L. E., McFarlane, D. C. & M. Preston Mullen, J. (1994) Integrality and separability of input devices. ACM Trans. Comput.-Hum. Interact. 1(1):3–26.

J.F.Kelley (2002) Kelley's personal web site: http://www.musicman.net/oz.html. Electronic Resource.

Jin, Y. K., Choi, S., Chung, A., Myung, I., Lee, J., Kim, M. C. & Woo, J. (2004) Gia: design of a gesture-based interaction photo album. Personal Ubiquitous Comput. 8(3-4):227–233.

Joseph J. LaViola, J., Feliz, D. A., Keefe, D. F. & Zeleznik, R. C. (2001) Hands-free multi-scale navigation in virtual environments. In: Proceedings of the 2001 symposium on Interactive 3D graphics. ACM Press, pp. 9–15.

Karam, M. & Hare, J. (2004) igesture project web site. Electronic Resource.

Karam, M. & schraefel, m. c. (2005a) A study on the use of semaphoric gestures to support secondary task interactions. In: CHI '05: CHI '05 extended abstracts on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 1961–1964.

Karam, M. & schraefel, m. c. (2005b) A taxonomy of gestures in human computer interactions. Tech. rep., Electronics and Computer Science, University of Southampton.

Karam, M. & schraefel, m. c. (2006) Investigating user tolerance for errors in vision-enabled gesture-based interactions. In: Submitted for Review: Proceedings of the working conference on Advanced visual interfaces.

Keates, S. & Robinson, P. (1998) The use of gestures in multimodal input. In: Proceedings of the third international ACM conference on Assistive technologies. ACM Press, pp. 35–42.

Kettebekov, S. (2004) Exploiting prosodic structuring of coverbal gesticulation. In: Proceedings of the 6th international conference on Multimodal interfaces. ACM Press, pp. 105–112.

Kjeldsen, R. & Kender, J. (1996) Toward the use of gesture in traditional user interfaces. In: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96). IEEE Computer Society, p. 151.

Kobsa, A., Allgayer, J., Reddig, C., Reithinger, N., Schmauks, D., Harbusch, K. & Wahlster, W. (1986) Combining deictic gestures and natural language for referent identification. In: Proceedings of the 11th coference on Computational linguistics. Association for Computational Linguistics, pp. 356–361.

Konrad, T., Demirdjian, D. & Darrell, T. (2003) Gesture + play: full-body interaction for virtual environments. In: CHI '03 extended abstracts on Human factors in computing systems. ACM Press, pp. 620–621.

Koons, D. B. & Sparrell, C. J. (1994) Iconic: speech and depictive gestures at the human-machine interface. In: Conference companion on Human factors in computing systems. ACM Press, pp. 453–454.

Kopp, S., Tepper, P. & Cassell, J. (2004) Towards integrated microplanning of language and iconic gesture for multimodal output. In: Proceedings of the 6th international conference on Multimodal interfaces. ACM Press, pp. 97–104.

Krueger, M. W., Gionfriddo, T. & Hinrichsen, K. (1985) Videoplace an artificial reality. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 35–40.

Krum, D. M., Omoteso, O., Ribarsky, W., Starner, T. & Hodges, L. F. (2002) Speech and gesture multimodal control of a whole earth 3d visualization environment. In: Proceedings of the symposium on Data Visualisation 2002. Eurographics Association, pp. 195–200.

Kuzuoka, H., Kosuge, T. & Tanaka, M. (1994) Gesturecam: a video communication system for sympathetic remote collaboration. In: Proceedings of the 1994 ACM conference on Computer supported cooperative work. ACM Press, pp. 35–43.

Latoschik, M. E. (2001) A gesture processing framework for multimodal interaction in virtual reality. In: AFRIGRAPH '01: Proceedings of the 1st international conference on Computer graphics, virtual reality and visualisation. ACM Press, New York, NY, USA, pp. 95–100.

Lee, C., Ghyme, S., Park, C. & Wohn, K. (1998) The control of avatar motion using hand gesture. In: Proceedings of the ACM symposium on Virtual reality software and technology. ACM Press, pp. 59–65.

Lenman, S., Bretzner, L. & Thuresson, B. (2002a) Computer vision based recognition of hand gestures for human-computer interaction. Tech. rep., Center for User Oriented ID Design.

Lenman, S., Bretzner, L. & Thuresson, B. (2002b) Using marking menus to develop command sets for computer vision based hand gesture interfaces. In: Proceedings of the second Nordic conference on Human-computer interaction. ACM Press, pp. 239–242.

Lumsden, J. & Brewster, S. (2003) A paradigm shift: alternative interaction techniques for use with mobile & wearable devices. In: Proceedings of the 2003 conference of the Centre for Advanced Studies conference on Collaborative research. IBM Press, pp. 197–210.

Maes, P., Darrell, T., Blumberg, B. & Pentland, A. (1997) The alive system: wireless, full-body interaction with autonomous agents. Multimedia Syst. 5(2):105–112.

McCrickard, D. & Chewar, C. (2005) Cognitive Systems: Human Cognitive Models in Systems Design., Lawrence Earlbaum, chap. Designing Attention-Centric Notification Systems: Five HCI Challenges, pp. 67 – 89.

McCrickard, D., Catrambone, R., Chewar, C. & Stasko, J. (2003a) Establishing trade-offs that leverage attention for utility: Empirically evaluating information display in notification systems. International Journal of Human-Computer Studies 8(5):547–582.

McCrickard, D. S. & Chewar, C. M. (2003) Attuning notification design to user goals and attention costs. Commun. ACM 46(3):67–72.

McCrickard, D. S., Chewar, C. M., Somervell, J. P. & Ndiwalana, A. (2003b) A model for notification systems evaluation assessing user goals for multitasking activity. ACM Trans. Comput.-Hum. Interact. 10(4):312–338.

McFarlane, D. C. (2002) Comparison of four primary methods for coordinating the interruption of people in human-computer interaction. Human-Computer Interaction 17:63–139.

Minsky, M. R. (1984) Manipulating simulated objects with real-world gestures using a force and position sensitive screen. In: Proceedings of the 11th annual conference on Computer graphics and interactive techniques. ACM Press, pp. 195–203.

Morency, L.-P., Sidner, C., Lee, C. & Darrell, T. (2005) Contextual recognition of head gestures. In: ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces. ACM Press, New York, NY, USA, pp. 18–24.

Morris, M. R., Huang, A., Paepcke, A. & Winograd, T. (2006) Cooperative gestures: multi-user gestural interactions for co-located groupware. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems. ACM Press, New York, NY, USA, pp. 1201–1210.

Moyle, M. & Cockburn, A. (2002) Gesture navigation: an alternative 'back' for the future. In: CHI '02: CHI '02 extended abstracts on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 822–823.

Myers, B. A. (1998) A brief history of human-computer interaction technology. interactions 5(2):44–54.

Myers, M. D. (1997) Qualitative research in information systems. MIS Quarterly 21(2):241–242.

Newman, W. M. (1997) Better or just different? on the benefits of designing interactive systems in terms of critical parameters. In: DIS '97: Proceedings of the conference on Designing interactive systems. ACM Press, New York, NY, USA, pp. 239–245.

Nickel, K. & Stiefelhagen, R. (2003) Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. In: ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces. ACM Press, New York, NY, USA, pp. 140–146.

Nishino, H., Utsumiya, K., Kuraoka, D., Yoshioka, K. & Korida, K. (1997) Interactive two-handed gesture interface in 3d virtual environments. In: Proceedings of the ACM symposium on Virtual reality software and technology. ACM Press, pp. 1–8.

Nishino, H., Utsumiya, K. & Korida, K. (1998) 3d object modeling using spatial and pictographic gestures. In: Proceedings of the ACM symposium on Virtual reality software and technology. ACM Press, pp. 51–58.

Osawa, N., Asai, K. & Sugimoto, Y. Y. (2000) Immersive graph navigation using direct manipulation and gestures. In: Proceedings of the ACM symposium on Virtual reality software and technology. ACM Press, pp. 147–152.

Ou, J., Fussell, S. R., Chen, X., Setlock, L. D. & Yang, J. (2003) Gestural communication over video stream: supporting multimodal interaction for remote collaborative physical tasks. In: Proceedings of the 5th international conference on Multimodal interfaces. ACM Press, pp. 242–249.

Ou, J., Oh, L. M., Fussell, S. R., Blum, T. & Yang, J. (2005) Analyzing and predicting focus of attention in remote collaborative tasks. In: ICMI '05: Proceedings of the 7th international conference on Multimodal interfaces. ACM Press, New York, NY, USA, pp. 116–123.

Oviatt, S., Cohen, P., Wu, L., Vergo, J., Duncan, L., Suhm, B., Bers, J., Holzman, T., Winograd, T., Landay, J., Larson, J. & Ferro, D. (2000) Designing the user interface for multimodel speech and pen-based gesture applications: State of theart systems and future research directions. Human-Computer Interaction 15:263–322.

Paiva, A., Andersson, G., Hk, K., Mourao, D., Costa, M. & Martinho, C. (2002) Sentoy in fantasya: Designing an affective sympathetic interface to a computer game. Personal Ubiquitous Comput. 6(5-6):378–389.

Paradiso, J. A. (2003) Tracking contact and free gesture across large interactive surfaces. Commun. ACM 46(7):62–69.

Paradiso, J. A., Hsiao, K., Strickon, J., Lifton, J. & Adler, A. (2000a) Sensor systems for interactive surfaces. IBM Syst. J. 39(3-4):892–914.

Paradiso, J. A., Hsiao, K. Y. & Benbasat, A. (2000b) Interfacing to the foot: apparatus and applications. In: CHI '00: CHI '00 extended abstracts on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 175–176.

Pastel, R. & Skalsky, N. (2004) Demonstrating information in simple gestures. In: Proceedings of the 9th international conference on Intelligent user interface. ACM Press, pp. 360–361.

Patel, S. N., Pierce, J. S. & Abowd, G. D. (2004) A gesture-based authentication scheme for untrusted public terminals. In: Proceedings of the 17th annual ACM symposium on User interface software and technology. ACM Press, pp. 157–160.

Pausch, R. & Williams, R. D. (1990) Tailor: creating custom user interfaces based on gesture. In: Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology. ACM Press, pp. 123–134.

Pavlovic, V. I., Sharma, R. & Huang, T. S. (1997) Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Trans. Pattern Anal. Mach. Intell. 19(7):677–695.

Peterson, L. R. & Peterson, M. (1959) Short-term retention of individual verbal items. J. Exp. Psych 58:193–198.

Pickering, C. A. (2005) Gesture recognition driver controls. IEE Journal of Computing and Control Engineering 16(1):27–40.

Pierce, J. S. & Pausch, R. (2002) Comparing voodoo dolls and homer: exploring the importance of feedback in virtual environments. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 105–112.

Pirhonen, A., Brewster, S. & Holguin, C. (2002) Gestural and audio metaphors as a means of control for mobile devices. In: CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, New York, NY, USA, pp. 291–298.

Quek, F., McNeill, D., Bryll, R., Duncan, S., Ma, X.-F., Kirbas, C., McCullough, K. E. & Ansari, R. (2002) Multimodal human discourse: gesture and speech. ACM Trans. Comput.-Hum. Interact. 9(3):171–193.

Quek, F. K. H. (1994) Toward a vision-based hand gesture interface. In: Proceedings of the conference on Virtual reality software and technology. World Scientific Publishing Co., Inc., pp. 17–31.

Reilly, R. B. (1998) Applications of face and gesture recognition for human-computer interaction. In: Proceedings of the sixth ACM international conference on Multimedia. ACM Press, pp. 20–27.

Rekimoto, J. (1997) Pick-and-drop: a direct manipulation technique for multiple computer environments. In: Proceedings of the 10th annual ACM symposium on User interface software and technology. ACM Press, pp. 31–39.

Rekimoto, J. (2002) Smartskin: an infrastructure for freehand manipulation on interactive surfaces. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 113–120.

Rekimoto, J., Ishizawa, T., Schwesig, C. & Oba, H. (2003) Presense: interaction techniques for finger sensing input devices. In: Proceedings of the 16th annual ACM symposium on User interface software and technology. ACM Press, pp. 203–212.

Rhyne, J. (1987) Dialogue management for gestural interfaces. SIGGRAPH Comput. Graph. 21(2):137–142.

Robbe, S. (1998) An empirical study of speech and gesture interaction: toward the definition of ergonomic design guidelines. In: CHI 98 conference summary on Human factors in computing systems. ACM Press, pp. 349–350.

Rodden, T., Crabtree, A., Hemmings, T., Koleva, B., Humble, J., kesson, K.-P. & Hansson, P. (2004) Between the dazzle of a new building and its eventual corpse: assembling the ubiquitous home. In: Proceedings of the 2004 conference on Designing interactive systems. ACM Press, pp. 71–80.

Roy, D. M., Panayi, M., Erenshteyn, R., Foulds, R. & Fawcus, R. (1994) Gestural human-machine interaction for people with severe speech and motor impairment due to cerebral palsy. In: Conference companion on Human factors in computing systems. ACM Press, pp. 313–314.

Rubine, D. (1991) Specifying gestures by example. In: SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques. ACM Press, New York, NY, USA, pp. 329–337.

Rubine, D. (1992) Combining gestures and direct manipulation. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 659–660.

Sagawa, H., Takeuchi, M. & Ohki, M. (1997) Description and recognition methods for sign language based on gesture components. In: Proceedings of the 2nd international conference on Intelligent user interfaces. ACM Press, pp. 97–104.

Sarker, S., Lau, F. & Sahay, S. (2001) Using an adapted grounded theory approach for inductive theory building about virtual team development. SIGMIS Database 32(1):38–56.

Schapira, E. & Sharma, R. (2001) Experimental evaluation of vision and speech based multimodal interfaces. In: Proceedings of the 2001 workshop on Percetive user interfaces. ACM Press, pp. 1–9.

Schiphorst, T., Lovell, R. & Jaffe, N. (2002) Using a gestural interface toolkit for tactile input to a dynamic virtual space. In: CHI '02 extended abstracts on Human factors in computing systems. ACM Press, pp. 754–755.

Schloss, P. J. & Smith, M. A. (1999) Conducting Research, Prentice Hall, chap. 5,7, pp. 53,89.

Schmandt, C., Kim, J., Lee, K., Vallejo, G. & Ackerman, M. (2002) Mediated voice communication via mobile ip. In: Proceedings of the 15th annual ACM symposium on User interface software and technology. ACM Press, pp. 141–150.

schraefel, m. c., Karam, M. & Zhao, S. (2003) mspace: interaction design for user-determined, adaptable domain exploration in hypermedia. In: In Proceedings of AH 2003: Workshop on Adaptive Hypermedia and Adaptive Web Based Systems. ACM Press, pp. 217–235.

Segen, J. & Kumar, S. (1998a) Gesture vr: vision-based 3d hand interace for spatial interaction. In: Proceedings of the sixth ACM international conference on Multimedia. ACM Press, pp. 455–464.

Segen, J. & Kumar, S. (1998b) Video acquired gesture interfaces for the handicapped. In: Proceedings of the sixth ACM international conference on Multimedia. ACM Press, pp. 45–48.

Sharma, R., Huang, T. S., Pavovic, V. I., Zhao, Y., Lo, Z., Chu, S., Schulten, K., Dalke, A., Phillips, J., Zeller, M. & Humphrey, W. (1996) Speech/gesture interface to a visual computing environment for molecular biologists. In: Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume III-Volume 7276. IEEE Computer Society, p. 964.

Silva, V. H. Z. & Arriaga, H. H. A. (2003) Evaluation of a visual interface for gesticulation recognition. In: Proceedings of the Latin American conference on Human-computer interaction. ACM Press, pp. 159–165.

Sinclair, P. A. S. & Martinez, K. (2002) Salt and pepper: Tangible hypermedia using the artoolkit. In: Poster in: Proceedings of the First IEEE International Augmented Reality Toolkit Workshop. ACM Press.

Smith, G. M. & schraefel, m. c. (2004) The radial scroll tool: scrolling support for stylus- or touch-based document navigation. In: Proceedings of the 17th annual ACM symposium on User interface software and technology. ACM Press, pp. 53–56.

Song, C. G., Kwak, N. J. & Jeong, D. H. (2000) Developing an efficient technique of selection and manipulation in immersive v.e. In: Proceedings of the ACM symposium on Virtual reality software and technology. ACM Press, pp. 142–146.

Stotts, D., Smith, J. M. & Gyllstrom, K. (2004) Facespace: endo- and exo-spatial hypermedia in the transparent video facetop. In: Proceedings of the fifteenth ACM conference on Hypertext & hypermedia. ACM Press, pp. 48–57.

Straub, Detmar, Gefen, D. & Boudreau, M.-C. (2004) The isworld quantitative, positivist research methods website. Electronic Source.

Strauss, A. & Corbin, J. M. (1998) Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory, Sage Publications, Inc.

Streitz, N. A., Geiler, J., Holmer, T., Konomi, S., Mller-Tomfelde, C., Reischl, W., Rexroth, P., Seitz, P. & Steinmetz, R. (1999) i-land: an interactive landscape for creativity and innovation. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 120–127.

Sturman, D. J. & Zeltzer, D. (1994) A survey of glove-based input. IEEE Comput. Graph. Appl. 14(1):30–39.

Sturman, D. J., Zeltzer, D. & Pieper, S. (1989) Hands-on interaction with virtual environments. In: Proceedings of the 2nd annual ACM SIGGRAPH symposium on User interface software and technology. ACM Press, pp. 19–24.

Sutherland, I. E. (1963) Sketchpad: A man-machine graphical communication system. In: Proceedings of the AFIPS Spring Joint Computer Conference 23. pp. 329–346.

Swindells, C., Inkpen, K. M., Dill, J. C. & Tory, M. (2002) That one there! pointing to establish device identity. In: Proceedings of the 15th annual ACM symposium on User interface software and technology. ACM Press, pp. 151–160.

Tang, J. C. & Leifer, L. J. (1988) A framework for understanding the workspace activity of design teams. In: CSCW '88: Proceedings of the 1988 ACM conference on Computer-supported cooperative work. ACM Press, New York, NY, USA, pp. 244–249.

Tarabella, L. & Bertini, G. (2000) Giving expression to multimedia performance. In: MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia. ACM Press, New York, NY, USA, pp. 35–38.

Teitelman, W. (1964) Real time recognition of hand-drawn characters. In: AFIPS Proceedings of Fall Joint Computer Conference. Spartan Books, pp. 559–575.

Thalmann, D. (2000) The virtual human as a multimodal interface. In: Proceedings of the working conference on Advanced visual interfaces. ACM Press, pp. 14–20.

Thayer, S. & Steenkiste, P. (2003) An architecture for the integration of physical and informational spaces. Personal and Ubiquitous Computing 7(2):82–90.

Tse, E., Histon, J., Scott, S. D. & Greenberg, S. (2004) Avoiding interference: how people use spatial separation and partitioning in sdg workspaces. In: CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work. ACM Press, New York, NY, USA, pp. 252–261.

Turk, M. (2004) Computer vision in the interface. Communications of the ACM 47(1):60–67.

von Hardenberg, C. & Berard, F. (2001) Bare-hand human-computer interaction. In: Proceedings of the ACM Workshop on Perceptive User Interfaces. ACM Press, pp. 113–120.

Ward, D. J., Blackwell, A. F. & MacKay, D. J. C. (2000) Dasher – a data entry interface using continuous gestures and language models. In: Proceedings of the 13th annual ACM symposium on User interface software and technology. ACM Press, pp. 129–137.

Weimer, D. & Ganapathy, S. K. (1989) A synthetic visual environment with hand gesturing and voice input. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press, pp. 235–240.

Weiser, M. (1993) Some computer science issues in ubiquitous computing. Commun. ACM 36(7):75–84.

Wellner, P. (1991) The digitaldesk calculator: tangible manipulation on a desk top display. In: Proceedings of the 4th annual ACM symposium on User interface software and technology. ACM Press, pp. 27–33.

Westeyn, T., Brashear, H., Atrash, A. & Starner, T. (2003) Georgia tech gesture toolkit: supporting experiments in gesture recognition. In: Proceedings of the 5th international conference on Multimodal interfaces. ACM Press, pp. 85–92.

Wexelblat, A. (1994) Natural gesture in virtual environments. In: Proceedings of the conference on Virtual reality software and technology. World Scientific Publishing Co., Inc., pp. 5–16.

Wexelblat, A. (1995) An approach to natural gesture in virtual environments. ACM Trans. Comput.-Hum. Interact. 2(3):179–200.

Wexelblat, A. (1998) Research challenges in gesture: Open issues and unsolved problems. In: Proceedings of the International Gesture Workshop on Gesture and Sign Language in Human-Computer Interaction. Springer-Verlag, pp. 1–11.

Wikipedia (2006a) Grounded theory. Electronic Source.

Wikipedia (2006b) Sketchpad. Electronic Resource.

Wild, P. J., Johnson, P. & Johnson, H. (2004) Towards a composite modelling approach for multitasking. In: TAMODIA '04: Proceedings of the 3rd annual conference on Task models and diagrams. ACM Press, New York, NY, USA, pp. 17–24.

Williams, K. (2000) pact of aviation highway-in-the-sky displays on pilot situation awareness.

Wilson, A. & Shafer, S. (2003) Xwand: Ui for intelligent spaces. In: Proceedings of the conference on Human factors in computing systems. ACM Press, pp. 545–552.

Wolf, C. G. & Rhyne, J. R. (1993) Gesturing with shared drawing tools. In: INTERACT '93 and CHI '93 conference companion on Human factors in computing systems. ACM Press, pp. 137–138.

Wu, M. & Balakrishnan, R. (2003) Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In: Proceedings of the 16th annual ACM symposium on User interface software and technology. ACM Press, pp. 193–202.

Zeleznik, R. & Forsberg, A. (1999) Unicam - 2d gestural camera controls for 3d environments. In: Proceedings of the 1999 symposium on Interactive 3D graphics. ACM Press, pp. 169–173.

Zhao, S. & Balakrishnan, R. (2004) Simple vs. compound mark hierarchical marking menus. In: UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology. ACM Press, New York, NY, USA, pp. 33–42.

Zimmerman, T. G., Lanier, J., Blanchard, C., Bryson, S. & Harvill, Y. (1987) A hand gesture interface device. In: Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface. ACM Press, pp. 189–192.

Zimmerman, T. G., Smith, J. R., Paradiso, J. A., Allport, D. & Gershenfeld, N. (1995) Applying electric field sensing to human-computer interfaces. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM Press/Addison-Wesley Publishing Co., pp. 280–287.