ViewPointer: Lightweight Calibration-Free Eye Tracking for Ubiquitous Handsfree Deixis

by

John David Smith

A thesis submitted to the
School of Computing
in conformity with the requirements for
the degree of Master of Science

Queen's University
Kingston, Ontario, Canada
September 2005

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-18811-8
Our file Notre référence
ISBN: 978-0-494-18811-8

Abstract

ViewPointer is a wearable eye contact sensor that detects deixis towards ubiquitous computers embedded in real world objects. ViewPointer consists of a small wearable camera no more obtrusive than a common Bluetooth headset. ViewPointer allows any real-world object to be augmented with eye contact sensing capabilities, simply by embedding a small infrared (IR) tag. The headset camera detects when a user is looking at an infrared tag by determining whether the reflection of the tag on the cornea of the user's eye appears sufficiently central to the pupil. ViewPointer not only allows any object to become an eye contact sensing appliance, it also allows identification of users and transmission of data to the user through the object. A novel encoding scheme is presented which is used to uniquely identify ViewPointer tags, as well as a method for transmitting URLs over tags. A number of scenarios of application are explored as well as an analysis of design principles. We conclude eye contact sensing input is best utilized to provide context to action.

Acknowledgments

The author would like to thank all the students in the Human Media Lab at Queens University for their help brainstorming ideas for this project. In particular, the author wishes to thank Daniel Cheng for his help developing the pupil and tag detection algorithms. In addition, the author wishes to thank Dr. Roel Vertegaal for his supervision and invaluable guidance throughout the duration of the project. But most of all the author wishes to thank his wife, Caroline Smith. Without her loving encouragement and support, this project would not have been possible.

Contents

List of Figures

Chapter 1

Introduction

Ubiquitous computing (sometimes referred to as pervasive computing) offers a new and emerging paradigm in the role of computers in our every day lives [37]. In the early days, computers were large mainframes accessed by many users at the same time. Later, the personal computer revolutionized the computing world by moving the computer away from centrally located mainframes and onto individual people's desk. Ubiquitous computing suggests that the relationship between people and computers is now moving into a "third wave", where individual people will now interact with many computers rather than a single box residing on a desk. This notion is different than the ubiquity of cell phones, PDAs, etc. where the computer is reduced to a small, portable, but distinct device. Rather, the ubiquitous computing paradigm suggests that the computer will disappear entirely into the background of our lives. In this way, computers can be thought of as becoming "invisible". By disappearing the computer into the natural environment, users can interact with the computer in a manner that more closely resembles the way users interact with the natural world.

A major problem with the trend toward ubiquitous computing is with the disappearance of the traditional input device. The computer must now sense the context of the user, not just respond to users' requests for actuation. This notion is called context aware computing. Current issues in context aware computing input include:

1. What is the nature of context? What does it mean for a computer to sense the context of a user, and how do different contextual measures combine into action? For example, if a user is proximous to a computing system, how does that information interact with other information about the user, such as her identity?

2. How does the user induce action? According to Norman [22], "How it works" knowledge plays an important role in understanding and anticipating the relationship between user activity and system action. When there are many hidden parameters that underlie context-aware input, it may be difficult for users to acquire the correct task action mappings, leading to unexpected system behavior.

3. How do I manipulate a system that is remote and invisible? What is the equivalent of a mouse for the real world? The question of how to point, or perhaps more appropriately, perform deixis [17] towards everyday computing objects is currently unresolved. Manual pointing ties up the hands, and can be slow and jittery due to dependence on the elbow and wrist joints. The most prevalent solution today is the unified remote. However, these tend to be overly complicated as they are incapable of performing deixis.

This thesis explores the use of a novel wearable eye pointing device that may address some of the above issues. Eye input provides a very specific kind of contextual information about the user: the objects that are subject to his or her visual attention [34]. Users' eye fixations tend to pertain to distinct objects, not arbitrary spatial coordinates. This led us to explore the notion of deixis as a metaphor for remote control. Rather than providing a spatial coordinate, deixis specifies the referent, for example, that object, in a spatial context. The act of looking at an object thus informs other forms of input activity, such as a button click or speech command, as pertaining to the semantics provided by that object.

There are a number of reasons why the use of eye input for remote control of everyday things is compelling [30]. In scenarios where the hands are busy or otherwise unavailable, the eyes provide an extra and independent channel of input. The eyes also move faster than any other body part. Users tend to already be looking at a target before they initiate manual action [14], and can produce thousands of eye movements without any apparent fatigue. User are also very familiar with the use of their eyes as a means for selecting the target of their commands, as they use eye contact to regulate their communications with others [35].

1.1 The Attentive User Interface

The Attentive User Interface (AUI) [28] paradigm attempts to address some of the problems of context aware computing by making the computer more aware of the user's action. With this knowledge, computers can then better manage the user's attention as a finite resource. As an example, consider a cellphone. A common cell phone has no awareness about the state of its user. When a call comes in, the phone will ring regardless of whether the user is in the middle of an activity that is of greater importance than the incoming call. The AUI paradigm suggests that awareness about the user's current attentional state could help the cell phone weigh the importance of the phone call against the user's current activity and decide how best to notify the user.

AUI models communication between a user and groups of ubiquitous computers after the way humans communicate in crowded social settings. Human beings regulate group communication by treating attention as a limited and competitive resource. More specifically, their brains only have the capacity to listen to a single speaker at a time. Humans manage this resource through a well developed turn taking mechanism [36]. To facilitate turn taking, 8 nonverbal cues are used [30]. However, in group conversations only one of these cues performs actual deixis, indicating the person to whom the speaker may be yielding the floor: eye contact [36]. The AUI paradigm utilizes sensing of eye contact to allow users to manage more effectively their opening and closing of communications with ubiquitous computing systems.

The Attentive TV is a system that illustrates the AUI communication model [30] and can be seen in Figure 1.1. The device begins to play video only when the device is receiving visual attention. When the user looks away, the device pauses playback. This behavior is modelled after the turn taking mechanism humans use in group conversations and therefore allows the user to use familiar non-verbal cues to control the device.

To facilitate this interaction, the Attentive TV has an integrated Eye Contact Sensor, which reports when the system is receiving eye contact [30]. The Eye Contact Sensor is similar to previous eye tracking technologies, but reports on the subject of the user's visual attention, rather then the location of the user's visual attention

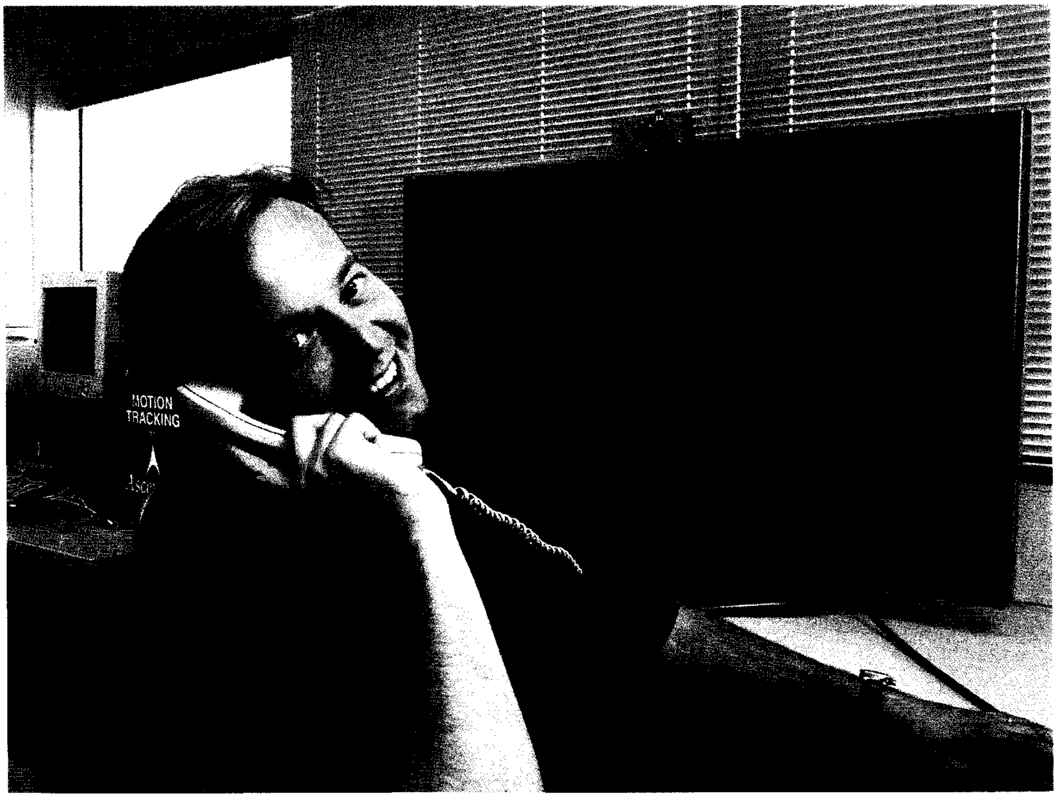

Figure 1.1: The Attentive TV pauses the video when it is not receiving visual attention [30].

within a given coordinate system [30]. In short, the eye contact sensor reports on "what" the user is looking at, while previous systems report on "where". We call the type of information attained from the eye contact sensor "deixis".

1.2 ViewPointer: Deixis with the Eyes

ViewPointer is a new eye contact sensor that is not only wearable, but also provides deixis towards computing objects in the real world, without calibration. By doing so, ViewPointer reports on "what" the user is looking at, rather than "where" in the visual scene they are looking. Also, ViewPointer can extend the reach of the computer towards objects in the real world that otherwise have no other computational component. This can be used to extend the AUI paradigm towards everyday objects, allowing virtually anything to become augmented with attentional input. In this sense, ViewPointer offers a step towards Weiser's vision of ubiquitous computing.

Chapter 2

The Human Visual System

The Human Visual System (HVS) is an extraordinarily complex system which is among the least understood systems in the human body. The HVS is what allows us to take two-dimensional images acquired from a stereo camera system, our eyes, and turn them into a 3D world full of discrete, moving objects. This section presents an overview of the human eye and human visual attention, two of the main components of the HVS that are the subject of current trends in eye tracking research and attentive user interface design.

2.1 The Human Eye: The World's Worst Camera

The human eye is a complex, adaptive biological system which has developed the nickname "the world's worst camera" [6]. It functions in a similar fashion to a digital camera, where the cornea, lens, iris, and pupil all work to draw light onto the retina. There, photosensitive cells transform the light into an electrical signal sent to the brain along the optic nerve. A diagram of the eye can be seen in Figure 2.1.

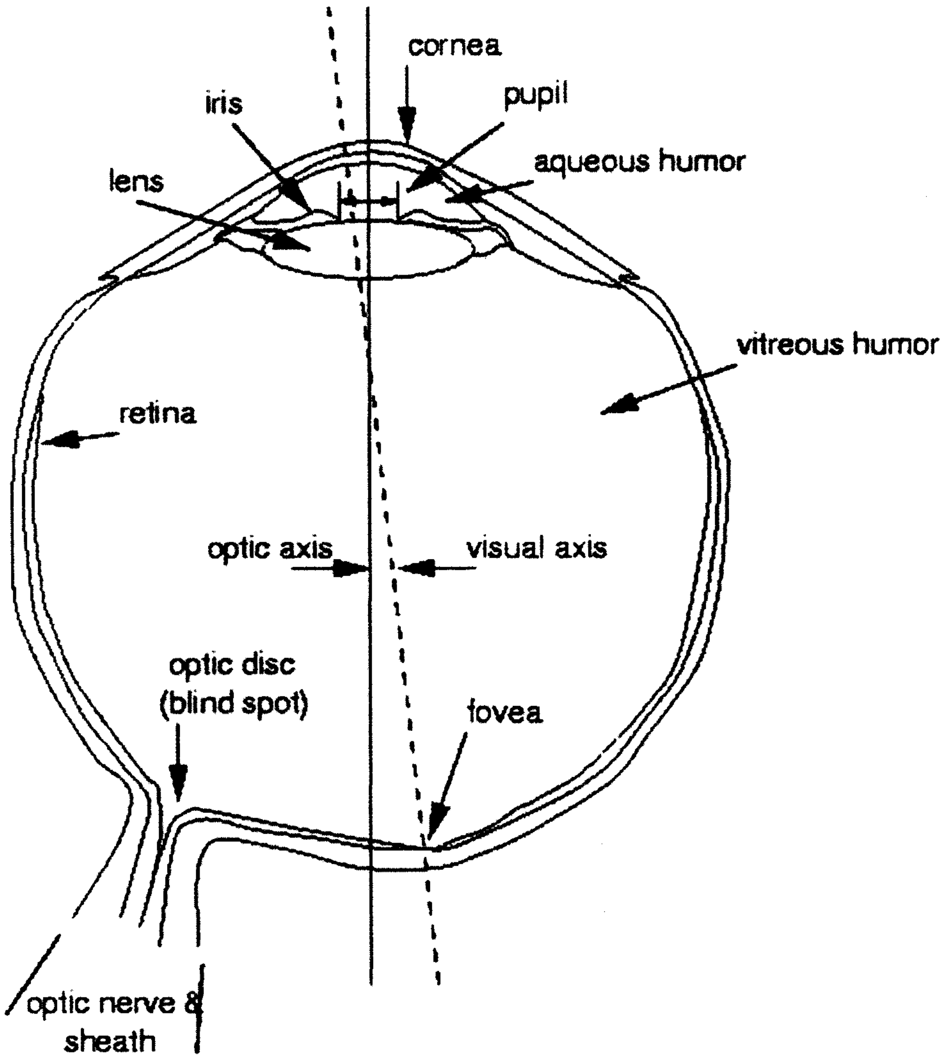

Figure 2.1: A diagram of the human eye. Adapted from [6].

2.1.1 The Cornea

The cornea is the large, transparent refractive layer that covers the front of the eye. Together with the crystalline lens, the cornea works to focus incoming light onto the retina. The cornea performs most of the refraction of incoming light but has no means of adjusting its curvature, and subsequently the amount of refraction performed by the cornea is fixed. In addition to its role in visual perception, the cornea also serves to protect the eye. The cornea contains sensitive nerve endings which cause an involuntary blinking reaction when touched [6].

2.1.2 The Pupil and the Iris

The pupil is the opening through which the cornea refracts light towards the retina. The pupil is clearly visible on the front of the eye as a black ellipse because most of the light that enters the pupil is absorbed by the tissue on the inside of the eye. The size and shape of the pupil vary widely across different species, mostly dependent on the visual requirements and lighting conditions the species typically encounters. The iris is the colored ring of muscles that surrounds the pupil. The iris controls the amount of light that reaches the retina by regulating the size of the pupil through an involuntary contraction called the "pupillary reflex" [6].

2.1.3 The Crystalline Lens

The crystalline lens (sometimes called simply the lens) resides behind the pupil and, along with the cornea, refracts light onto the retina. The crystalline lens is flexible and its curvature is controlled by ciliary muscles. By changing the curvature of the lens, the eye can change its focus among objects at different distances. In this way, the cornea can be thought of as the "coarse" lens and the crystalline lens can be thought of as "fine" [6].

2.1.4 The Retina

The retina is a thin layer of cells at the back of the eye that converts light into electrical signals. The retina is the only part of the human brain that is externally visible. The photoreceptor cells in the retina are of two types: rods and cones. Rods detect the intensity of light, but do not detect frequency (color). Rods are used for low-intensity, low-resolution vision and are useful for night vision. Cones are sensitive to high intensity light and can detect color. A high concentration of cones can be found in a small spot (about 0.01% of the visual field) called the fovea. The fovea provides sharp, high-resolution color vision in about 2 degrees of visual angle in the center of our visual field. The peripheral regions of the retina contain few cones but a higher density of rods [6].

2.2 Human Visual Attention

The study of the concept of human visual attention extends back into the late 19th century. Early studies relied on simple naked-eye observations of others or self introspection since the technology to take more scientific measurements of eye movements was not yet available. Later, eye tracking technology and advances in neuroscience have lead to an increased understanding of the low-level function of attention [6].

2.2.1 "What" versus "Where"

At the start of the 20th century, Hermann von Helmholtz published the first notable work on the subject of visual attention [2]. In this work, von Helmholtz observed that the detail in which the eyes can perceive an object decreases as the object moves further away from the focal point of our gaze. Because of this, the focal point of our gaze is typically moved in coordination with our visual attention. He also observed that we have an ability to focus our attention to objects in our visual periphery without necessarily making an eye movement towards that object. However, we have a natural tendency to make eye movements towards the space that is the focus of our attention so that we can perceive that space in more detail. In this way, it is said that Helmholtz was interested in the "where" of our visual attention, in that he observed eye movements in terms of their spatial locations within the visual field [6].

The famous psychologist William James countered Von Helmholtz's view of attention with an interpretation more closely related to higher-level mental processes such as imagination or anticipation [15]. James believed attention focused not on "where" the subject is looking, but rather "what" the subject is looking at. In this way, the focus of attention resides on the identity, meaning, or expectation of the object rather than its location in the visual field [6]. The contrast between Von Hemholtz's and James' views of visual attention has influenced contemporary views, roughly corresponding to the "foveal" (James) and "parafoveal" (Von Helmholtz) aspects of visual attention [6].

2.2.2 Attention as a "Selective Filter"

In the 1950's, Donald Broadbent hypothesized that attention serves as a means to manage a limited resource of sensory channels in the brain [1]. To test this theory, he performed a series of experiments on auditory attention. Subjects wore a headset that simultaneously presented a different series of integers in each ear (for example: 4-3-7 in the left, 9-3-1 in the right) and were asked to report what they heard. In every trial, subjects reported the two sets of integers serially (in our example: 4-3-7-9-3-1, or 9-3-1-4-3-7) rather than interweaving the integers. Broadbent concluded that this phenomenon suggested a two-stage method of processing input. The first stage has a high capacity offering low-level parallel processing. The second stage interprets the input further, but is a limited resource that offers only serial processing. He surmised that attention was the filter that selectively manages the second stage of input processing.

J. Anthony Deutsch and Diana Deutsch rejected the notion that attention serves as a selective filter [5]. They theorized that the selective filter necessary to reduce the amount of input that is processed by the second stage would have to be at least as complex as the second processing step itself. Therefore, they believed that a preset collection of "importance weightings" is used to filter input after all processing is completed. In this way, the observable attentional effects are a function of importance of information rather than the information itself [6].

2.2.3 Scanpaths

In the 1960's, Yarbus performed experiments that challenged the Gestalt view of visual object recognition [38]. In these experiments, Yarbus gave subjects questions related to an image, and then recorded their eye movements as they examined the image. In this study, Yarbus found that the eyes follow a sequential viewing pattern and foveate over various regions of interest in a scene, and that the viewing pattern was different depending on the leading question provided to the subject.

Later, Noton and Stark performed a similar study, but did not give subjects leading questions before presenting the image [23]. The record of the eye movements on the image was termed a "scanpath". Their work extended Yarbus' work in that it demonstrated that visual patterns tend to fixate on interesting regions even when no question about the image is provided. In addition, they found the path between interesting regions followed no predictable sequence.

Yarbus and Noton and Stark's work challenged the traditional Gestalt view that recognition is a one-step process which can be performed in parallel. Instead, their work suggested that recognition is a serial process that assembles a collection of regions of interest which are acquired in no particular order by the eyes. Also, this work supports James' notion that attention is interested in "what" the subject is looking at, where the object of the subject's attention is the region of interest in the image [6].

2.2.4 Attention as a "Spotlight"

Following the findings of Yarbus and Noton and Stark, Posner suggested that attention serves as a kind of "spotlight" that moves about the visual field [26]. Posner felt this spotlight was independent of eye movements, and instead was an attentional mechanism having limited spatial size, which is supported by Noton and Stark's findings of serial foveated attention [6]. He suggested visual attention has two components: orienting and detecting. He described the orienting component as a mental task that is performed in parallel, can operate independent of the eyes, and must precede the detecting component. The detecting component is described as being context-sensitive and must be attached to the input signal.

2.2.5 Treisman's Feature Integration Theory

An important contribution of Posner's work is the dissociation of attention from low-level foveal vision. To this end, James' "what" of attention can be thought of as Posner's orienting component and correlates to serial foveal vision. Similarly, Von Hemholtz's "where" of attention can be thought of as Posner's detecting component, and correlates to parafoveal processing, which determines the next focus of attention and operates in parallel [6]. Treisman brought these two components together with the Feature Integration Theory [33]. Here, the "where" of attention is used to create a location-based map of features present in the visual field, but makes no claim about what those features are. Later, the "what" of attention selects features of this map and "glues" them together into discrete objects [6].

2.2.6 The Attentional "Window"

Kosslyn offered an alternative view of Treisman's attenuation filter [16]. In Kosslyn's model, attention is a "window" that is used for pattern detection inside a "visual buffer". Because the contents of the entire visual buffer are too large to pass downstream, the "window" serves as a filter similar to Broadbent's selective filter. The window is not of a fixed size, and can be adjusted to accommodate complex scenes.

Kosslyn's model also fits well with the concept of "mental imagery", which is the brain's ability to mentally construct a scene or experience that was not induced by an external stimulus [6].

Chapter 3

Literature Review: Eye Tracking

Eye tracking is a technique used in a variety of areas, including neuroscience, psychology, cognitive science, human computer interaction, etc [6]. However, most previous usage has focused on using the eye tracker as a tool for studying the cognitive processes of the brain, or the human visual system itself. The concept of using an eye tracker as an input device for the control of a computer is a much less studied area of research, mostly focusing on helping those with motor impairments such as quadriplegics, where the eyes could be used as a substitute for the hands [6].

3.1 Eye Tracking Methodology

The simplest approach to eye tracking is with electrodes placed on the user's skin around the eye. Muscle activity is monitored to track the position of the eye relative to the head. However the electrodes necessary for this approach are somewhat invasive, and this method is mostly useful for measuring the position of the eye relative to the head rather than to an external object such as a computer monitor [6].

A more invasive technique involves placing a contact lens with a magnetic coil against the user's cornea and adhering it in place with suction. The coil then moves with the eye through an electromagnetic field, which produces an induced current. This current is then measured to determine the location of the user's eye [6].

Today, most eye trackers are video-based corneal reflection systems that monitor the user's eye with a camera, either remotely or worn on the user's head. A light source is placed such that its reflection on the cornea is visible to the camera. Because the surface of the cornea is roughly spherical, the position of this reflection will remain stationary with the user's head movements. The eye is then tracked by following the user's pupil as it moves relative to this reflection. By tracking two components of the eye (the pupil and the corneal reflection), head movements (two components moving together) can be distinguished from eye movements (one component moving relative to the other) and therefore the head is allowed to move freely within the visual range of the camera [6].

3.2 Early Corneal Reflection Eye Trackers

The first known corneal reflection eye tracker was the famous Oculometer system built for the United States Air Force by Merchant and Morrissette [20]. In this work, Merchant and Morissete developed many of the low-level mathematical details necessary to produce a gaze coordinate from an image of the eye. They developed a root-mean-square algorithm that is used to calibrate the tracking method to individual users. Also, they developed higher order polynomial equations necessary for correcting nonlinearities.

3.3 Desktop Eye Trackers

The most common form of eye tracking today is the desk-mounted tracker. These systems typically reside underneath the user's PC and track the users point of gaze on a standard monitor. They operate in real time and typically report the users gaze point in screen coordinates. Desktop trackers are highly accurate, with a precision of approximately 1 cm at a viewing distance of approximately 60 cm. However, existing systems are very expensive, limit users' head movements, and require calibration for individual users. Their vision is typically restricted to within 60 cm from a surface, making them unusable for most ubiquitous computing applications.

Commercial desktop eye tracking systems, such as those produced by Polhemus, LC Technologies (LCT), and Tobii Incorporated illuminate the eye with an infrared LED light source and monitor the eye with a digital camera. The following is a survey of current state-of-the-art desktop eye trackers:

3.3.1 Polhemus VisionTrak ETL-400

The Polhemus VisionTrak ETL-400 eye tracker, developed by ISCAN, tracks the user's eye by illuminating it with an off-axis LED infrared light source and a camera positioned beneath the user's monitor [12]. The illumination is invisible to the user. The system has a subpixel resolution of 1500 x 2200 pixels and true 60 Hz data updates. Because the camera is remote and requires a close-up view of the eye, an automatic pan/tilt system is used.

Because the scene is illuminated with an infrared LED, the camera is fitted with an infrared filter. This filter causes the image acquired by the camera to be grey scale. The pupil is revealed as a dark ellipse in the center of the user's iris. The camera also detects the reflection of the infrared light source on the user's cornea. The relationship between the user's pupil and this reflection are used to determine eye position. Gaze coordinates are reported relative to the user's monitor.

The ETL-400 system requires a calibration step to be completed to correlate eye position with screen coordinates. This step requires the user to look at either 5 or 9 points on the monitor. When properly calibrated, the system has an accuracy of better than one degree visual angle over a radius of 20-25 degrees of visual angle.

A major drawback to the VisionTrak system is setup. The computer vision algorithm used to track the pupil relies on user input to set the threshold between the pupil and the surrounding iris. This threshold must be set quite precisely and must be reset with changes in infrared lighting conditions. Problems with this setup causes a great deal of noise and data loss.

Upgrades are available to help increase the feature set of the VisionTrak system. Available upgrades include increased range of the eye tracker, greater point reporting speed (up to 240 Hz), binocular tracking, head tracking, and also multiple subject tracking.

3.3.2 Tobii 1750

The Tobii 1750 eye tracker [13] is a similar system to the VisionTrak ETL-400. The Tobii uses the same illumination and pupil detection technique, however all the hardware is integrated into a 17" TFT display (see Figure 3.3). By having the camera at a fixed position relative to the display, the Tobii is more tolerant to long periods between calibrations than the VisionTrak. The Tobii also uses a much more powerful array of IR LEDs to light the scene. Because the Tobii uses much brighter LEDs than

Figure 3.1: The Polhemus VisionTrak ETL-400 desktop eye tracker. Adapted from [12].

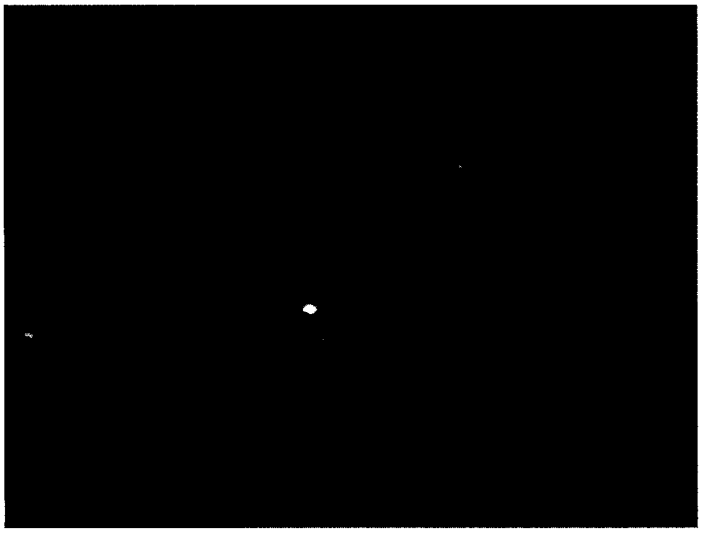

Figure 3.2: The eye illuminated with an off-axis LED. Adapted from [21].

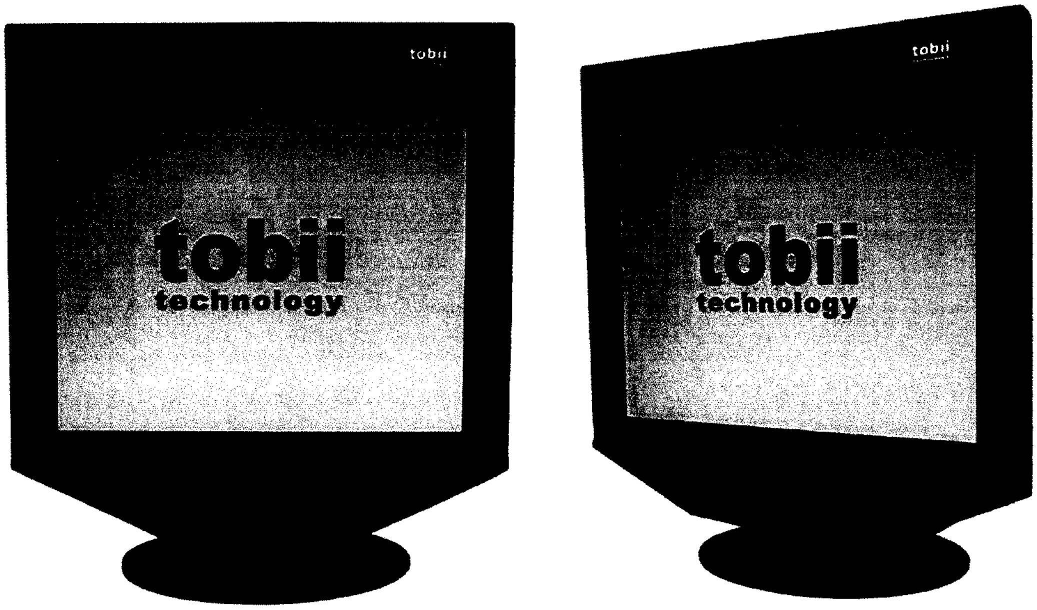

Figure 3.3: The Tobii 1750 Eye Tracker. This eye tracker is integrated into a 17" TFT display. The eye tracker uses a collection of high powered off-axis LEDs to light the user's eyes. Adapted from [13].

the VisionTrak, the Tobii is much less prone to the setup problems that come with setting the threshold values for the computer vision algorithm, and consequently is more reliable than the VisionTrak. Also, the camera has a sufficient viewing angle such that it does not need a pan/tilt system to track the user's eye.

The Tobii also has the added advantage of tracking both of the user's eyes. Because of this, the Tobii has an accuracy advantage over the VisionTrak. The Tobii is accurate to approximately 0.5 degrees of visual angle. The Tobii follows the same calibration technique as the VisionTrak, but also offers a 15 point calibration sequence in addition to the 5 and 9 point sequences.

3.3.3 LC Technologies Eye Gaze System

The LC Technologies Eye Gaze system (Figure 3.4) uses a similar setup as the Tobii and VisionTrak systems, but detects the pupil in a different way [11]. Rather than using off-axis LEDs to light the area around the pupil, leaving a distinct dark ellipse for the pupil, the LC illuminates the pupil itself. To do this, the LC takes advantage of the property held by most mammalian eyes that light which enters the eye typically leaves in the same direction in which it entered after reflecting off the photoreceptors on the retina. This property is most commonly seen as the "red eye" effect in flash photography.

To induce this effect, the LC places the LEDs used to light the eye in a ring around the camera lens. This placement causes the LEDs to shine down the visual axis of the camera rather than from off-axis angles which are used with the Tobii and VisionTrak. Because of this, the light from the LEDs enters the eye from the same direction as the camera and consequently leaves in the direction of the camera. The result is an illuminated pupil, as seen in Figure 3.5. Because of this, the computer vision algorithm used to detect the user's pupil is significantly easier to configure than the VisionTrak's algorithm, and is more tolerant to changes in lighting conditions.

3.3.4 Pupil Cam

As part of the Blue Eyes project [21], researchers at IBM Almaden developed a lightweight eye tracker called Pupil Cam. This system uses both a bright pupil and a dark pupil to track the user's point of gaze [32]. The system alternates flashing on- and off-axis LEDs at a rate coordinated with the camera's frame rate. The result is such that the frames retrieved from the camera have alternating bright and dark

Figure 3.4: The LC Technologies Eye Gaze System. Adapted from [11].

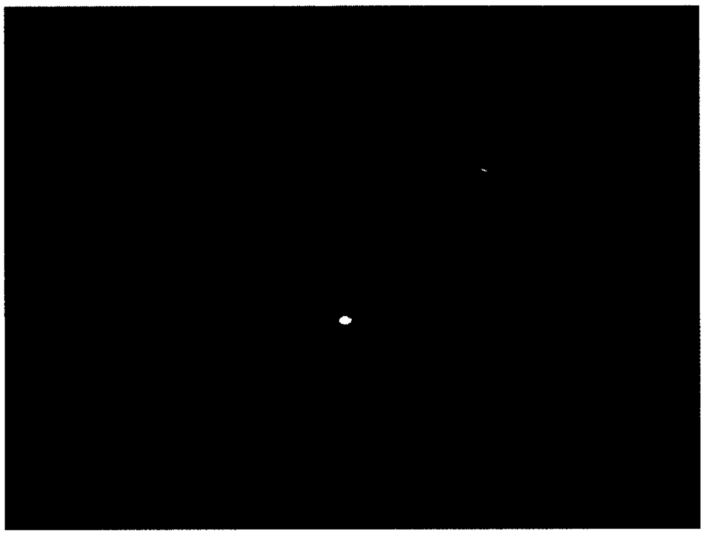

Figure 3.5: The pupil illuminated with an on-axis LED. Adapted from [21].

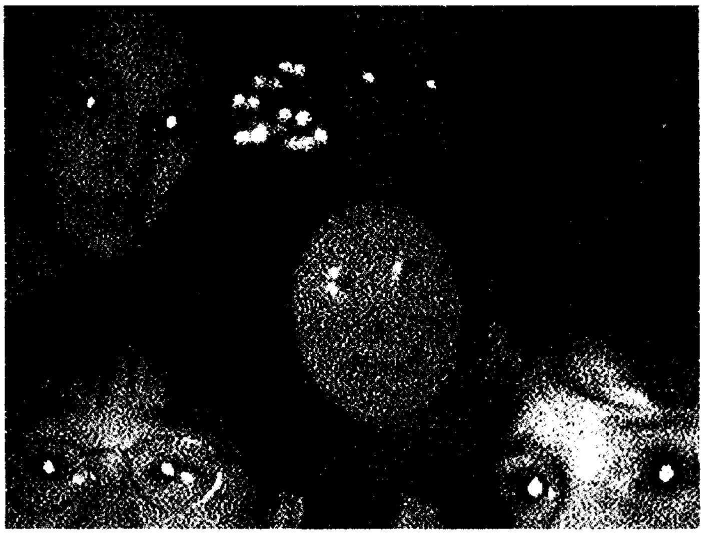

Figure 3.6: The pupils illuminated with an on-axis LED. Adapted from [21].

pupils. The two images are then subtracted, leaving only the user's pupils. This technique makes the Pupil Cam much easier to set up than previous eye trackers since the image processing technique used to acquire the pupils is more robust and requires less tuning.

3.4 Head-Mounted Eye Trackers

Recent head-mounted eye trackers such as the one seen in Figure 3.9 work in a similar fashion to common desktop eye trackers. Rather than report the coordinates of the user's gaze point on a screen, these systems typically report the user's gaze point relative to the image obtained from a second camera which is fixed to the user's head.

Current head-mounted eye trackers have some major drawbacks. These systems are generally bulky and expensive ($20,000+, see Figure 3.9). Calibration requires them to be affixed to the head, since otherwise the gaze point would shift with shifts

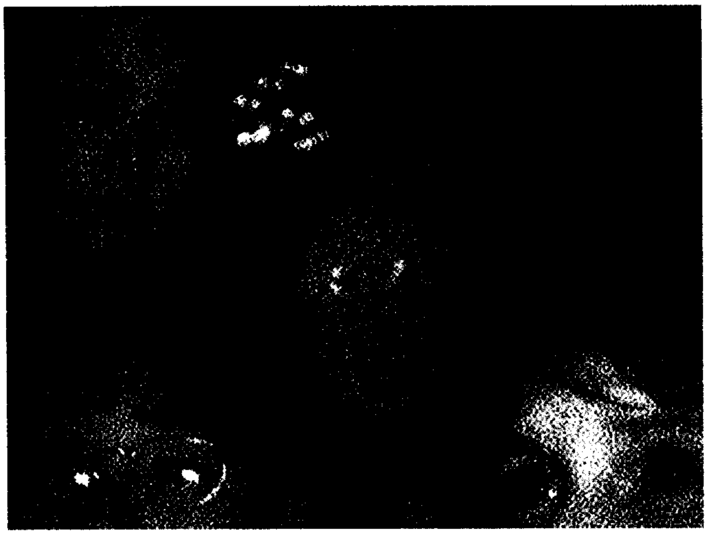

Figure 3.7: The pupils illuminated with an off-axis LED. Adapted from [21].

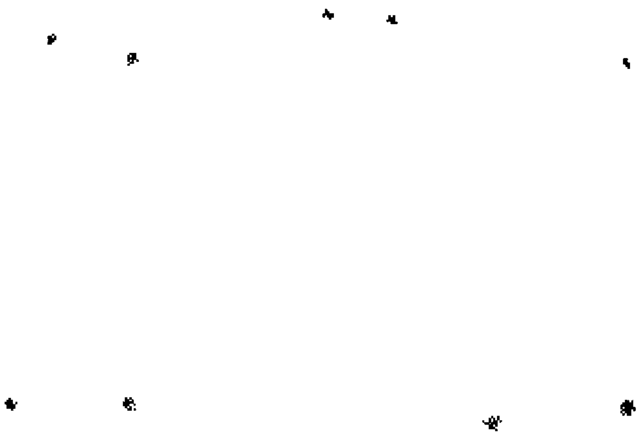

Figure 3.8: The two images subtracted, leaving only the user's pupils. Adapted from [21].

of the tracker relative to the head. Moreover, most headmounted trackers measure gaze points relative to the scene camera fixed to the head, rather than the scene itself. This means they cannot intrinsically detect which real-world object the user is looking at.

3.5 The Eye Contact Sensor

Although there is a long history of research into eye tracking technologies, the detection of eye contact with devices is a relatively novel area of research. The term eye contact is somewhat misleading, as the device typically does not reciprocate gaze [3]. Instead this term is used to indicate the use of the eyes for deixis towards an object [28].

Eye contact sensing reports on visual attention in a different way than traditional eye tracking. Recalling Von Hemholtz's studies of visual attention, traditional eye tracking reports on the "where" of visual attention, that is a coordinate relative to some point in the user's visual field. Eye contact sensing reports on what object the user is looking at, which more closely resembles James' "what" of visual attention. This difference make eye contact sensing a more useful tool when designing user interfaces that fit the AUI paradigm.

Yu and Ballard [39] report a system that uses computer vision from a head-mounted scene camera to determine the object the user is fixating at. While this approach is promising, it is also cumbersome as it relies heavily on prior knowledge of the appearance of objects in the visual scene. A recent development is that of the Eye Contact Sensor (ECS) [30], shown in Figure 3.10. The ECS is designed to serve as an augmentation for the object that is to report visual attention. This technology

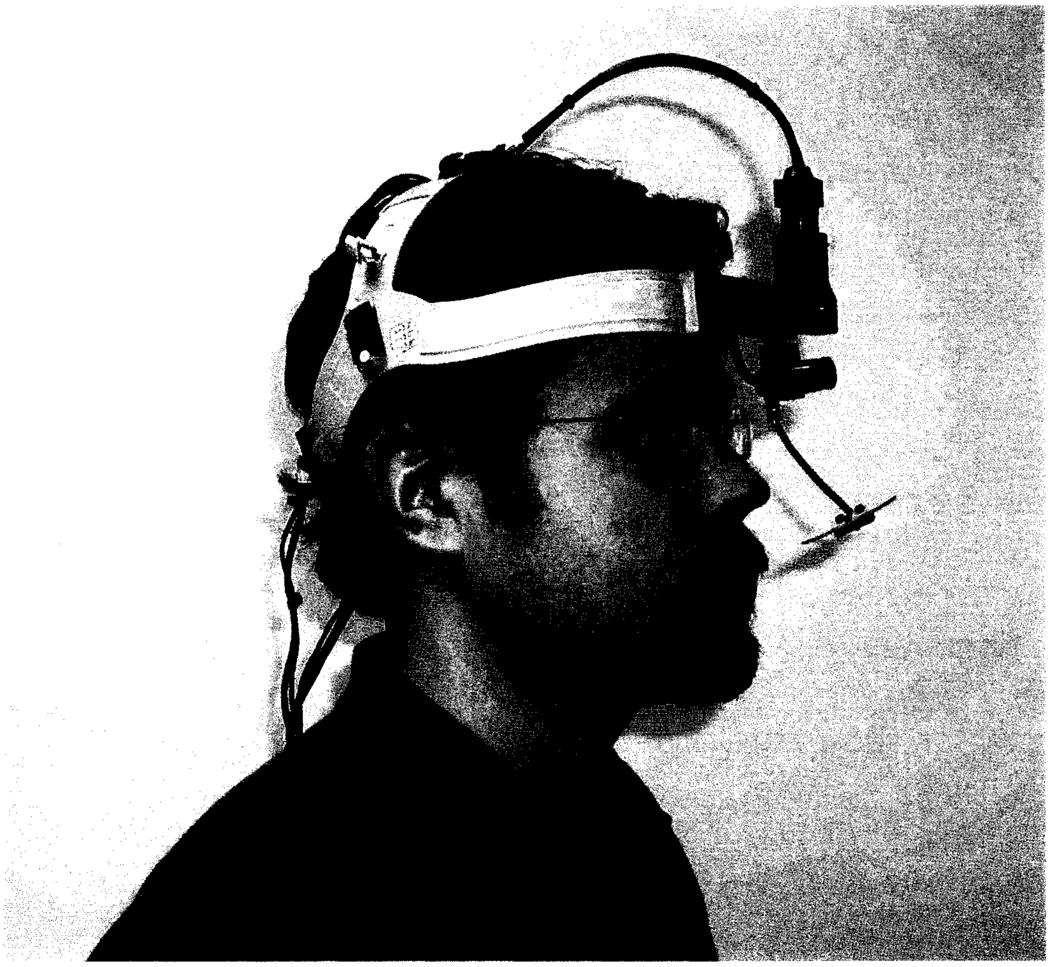

Figure 3.9: ASL Headmounted Eye Tracker. This eye tracker requires tight mounting on the head to avoid shifts in calibration accuracy. It does not report coordinates relative to the scene.

Figure 3.10: Eye Contact Sensor (Rev 1). Detects eye contact with an accuracy of about 9 degrees at distances up to 1 m.

applies the same computer vision techniques used with IBM's pupil cam to locate the user's pupil, but do not relate the location of the user's gaze to a coordinate system, eliminating the need for calibration. In addition, the system can be produced for a fraction of the cost of a commercial eye tracker. The ECS not only detects the users pupil, but also detects a corneal reflection of an LED. When this glint is in the center of the users pupil, the sensor determines eye contact is made. The sensor is small, works in real time and is well suited for ubiquitous computing scenarios. New MegaPixel versions can detect eye contact at up to 2-3 meters distance [30].

3.6 Head-Mounted Eye Contact Sensors

Head-Mounted Eye Contact Sensors have been proposed to detect eye contact between humans. Two such systems are known to exist:

3.6.1 Eye-R

The first known head-mounted eye contact sensor is Eye-R [27]. The system was designed to be used as an augmentation for any common pair of glasses that detects eye contact with other wearers in the environment. The system contains an IR transmitter and receiver (Figure 3.11) pointed into the user's environment. This transmitter is fitted with an IR LED with a narrow angle of transmission (17 - 20 degrees) that transmits a unique code. This allows the system to determine when the user's head is oriented towards another user.

Fluctuations in IR light are used to detect the user's pupil. An IR LED and a phototransistor is pointed inward towards the user's eye. As the user's eye moves, the amount of IR reflected from the eye changes. A fixation is detected when the amount of IR reflected from the wearer's eye remains constant. Eye contact is determined when the user's head is oriented towards another user and the user's eye is fixated. The system communicates with a base-station with a wireless transceiver that is connected to a PC with a serial port. A major advantage to this approach is detection speed. A sample-and-hold circuit at 60 Hz is used to detect eye contact, and an onboard PIC micro-controller is used to detect fixations within the signal. However Eye-R assumes that the user's gaze direction is always in the direction the head is oriented, and does not actually track the direction of the eye itself.

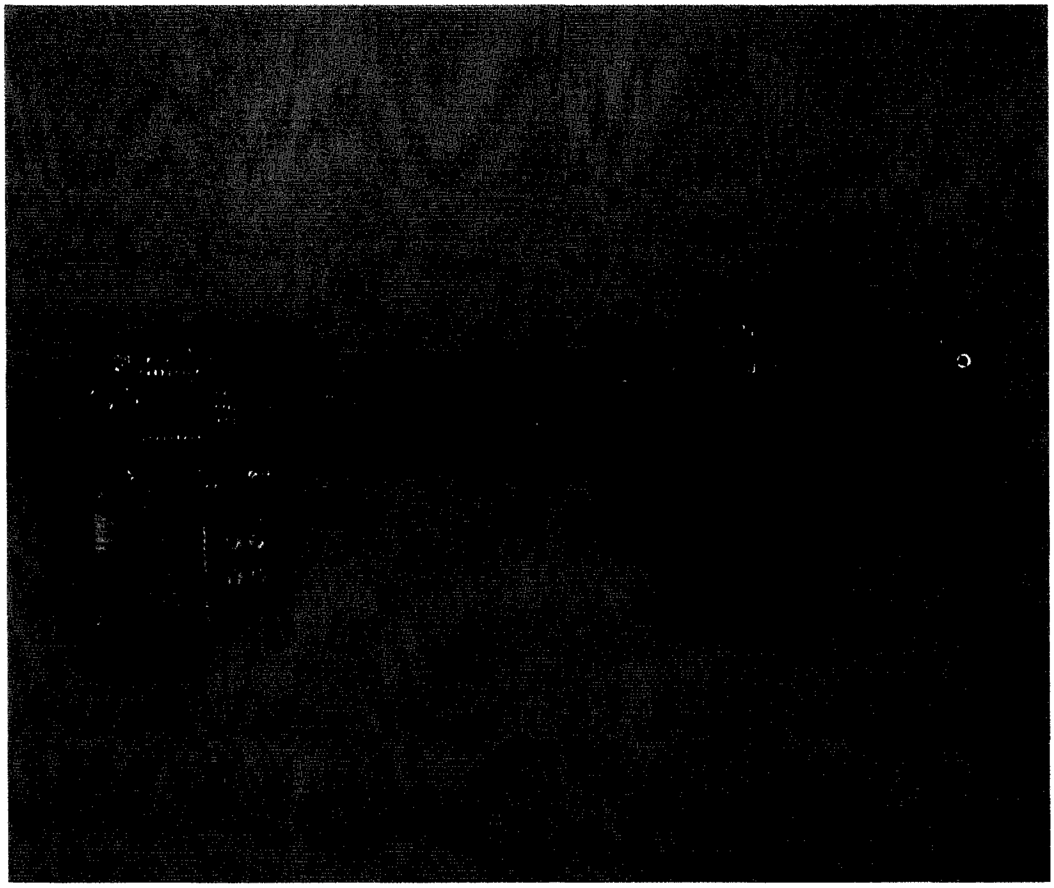

Figure 3.11: The Eye-R headset augments an ordinary pair of glasses, and reports on when two users are making eye contact [27].

3.6.2 ECSGlasses

ECSGlasses (Figure 3.12) are another system designed to detect eye contact between people [29]. This system uses the same eye contact sensing technique as the Eye Contact Sensor, but the camera is mounted between the wearer's eyes. This allows the system to detect eye contact with the wearer without requiring the other person to wear ECSGlasses.

The system was designed to report on the status of the user's attention. The Attentive Cell Phone uses ECSGlasses to mediate interruptions received from a ringing cell phone based on the user's attentive context [29]. Similarly, the Attentive Messaging Service reports the user's attentive state to others by setting the status on a common instant messaging application. The Attentive Hit Counter uses ECSGlasses to maintain a record of how much eye contact a user receives. Finally, eyeBlog is a system that records an optimized visual record of all the conversations a user participates in [29]. The video is recorded from the vantage point of the wearer since the camera is mounted directly between the wearer's eyes. When the user is receiving eye contact, eyeBlog begins to record both the video received from the ECS camera and also audio recorded from an on board microphone. When the user is no longer receiving eye contact, eyeBlog stops recording [29].

3.7 Eye-Based Interaction: Pointing

The most obvious means of eye-based interaction is in pointing tasks, where the object the user is looking at is considered to be selected. Robert Jacob published the first notable work on this topic in 1990 [14]. In this work, the eyes were used largely

Figure 3.12: ECSGlasses detect when the wearer is receiving eye contact.

as a substitute for the mouse and an experiment was performed to evaluate this technique. The most interesting result from this work suggests that simply selecting whatever object the user is looking at is quite undesirable, as it leads to the "Midas Touch" effect. Instead, Jacob proposed looking behavior should be accompanied by a secondary actuation step, such as pushing a button or fixating of the object for a short time.

Other work has focused on using the eyes in more passive ways. These systems seek to reduce the amount of awareness about eye movements the user must have, but still use the eyes for control. Most commonly, the eyes are proposed as an auxillary input channel to the hands. Manual And Gaze Input Cascaded (MAGIC) pointing is one such example [40]. This technique uses the eyes to position the cursor roughly near the object the user is looking at. The hands can then be used to move the cursor to an object local to the user's visual attention. This allows the user to move the cursor long distances with his eye muscles, which are the fastest in the human body, but still retain the fine cursor control afforded by manual input with the mouse. Also, users reported that the "magical" aspect of the technique was that by passively following the user's visual attention, the cursor seemed to follow user intent.

3.7.1 Pointing in Virtual Reality

Eye movements have also been proposed as a modality for pointing within virtual environments [31]. These systems typically correlate the user's gaze into a vector defined by virtual world coordinates. Typically, 2D gaze coordinates are retrieved from the eye tracker and then projected into the world using simple ray casting.

Tanriverdi and Jacob proposed that eye movements could be used as an active pointing device for 3D object selection in virtual environments presented in a head-mounted display (HMD) [31]. In this work, the eye was tracked in 2D in screen coordinates in the HMD. Ray casting was used to select the nearest object rendered to the pixel residing at the gaze coordinate, and a dwell time was used to avoid the Midas Touch effect [14]. This work compared eye-based selection to the traditional use of the hands to reach out and touch an object to select it. The study revealed the eyes to be considerably faster than hands for this selection task. This work was later extended by Cournia et al. to include a comparison between gaze-based pointing and hand-based pointing where a ray-casting technique was used by the hands [4]. This study revealed that the eyes are not necessarily faster than the hands when the user does not have to reach to touch the target, and instead can point at objects with the hand from a distance.

3.7.2 Eye-based Interaction: Deixis

Maglio et al. at IBM Almaden performed a Wizard of Oz study to determine the effectiveness of speech as a tool for the control of a smart office [18]. Users were asked to perform a variety of tasks using only speech commands in a simulated office environment that contained a variety of familiar appliances. A between-subjects design was used to compare the difference between providing visual feedback that the command was successfully completed on a single specialized component (the authors compare this to HAL in 2001) or on the device receiving the speech command. Interestingly, the study revealed that users tend to look at the device that is to receive the speech command shortly before actually issuing the command, no matter where the feedback is to be presented. The authors suggest this behavior is an emulation of the way humans use eye contact to manage social situations.

Later, Oh et al. at MIT built upon the findings of Maglio et al. by developing Look-to-talk: a gaze-aware speech system that utilizes eye contact detection in an interactive way [25]. Interfaces that utilize Look-to-talk only respond to spoken commands when they are receiving visual attention. Since Maglio et al. demonstrated that users tend to look at devices before issuing spoken commands, it was thought that the user's gaze could serve as a natural estimation of the object to which the user intended to issue the command. This removes the need to issue a spoken word to select a device. So for example, the phrase "Turn on the lamp" could be shortened to "on" since the user is typically looking at the lamp when issuing the command.

To study the usefulness of such a system, Oh et al. performed two studies: one using a working implementation of the concept and one using the Wizard of Oz experiment technique [25]. In the working implementation, head orientation was used as an estimation for the direction of the user's gaze. The two studies compared the use of Look-to-talk to "Talk-to-talk", where the user speaks the name of the object that will receive the command, and "Push-to-talk", where the user pushes a button corresponding to the object that will receive the command. The studies found that users preferred using the Look-to-talk and Talk-to-talk over Push-to-talk when interacting with computing agents, and that when given the choice, most often used Look-to-talk.

3.7.3 EyePliances

Shell et al. [30] applied the principle of Look-to-Talk when developing EyePliances: standard home appliances augmented with Eye Contact Sensors that respond to the users visual attention. Each EyePliance is augmented with its own eye contact sensor. The user signals attention to an EyePliance by looking at it, which is typically used to open up a communication channel with the corresponding appliance. A user can then interact with the EyePliance using speech, remote, or manual controls. EyePliances allow the user to interact with many devices in a way comparable to how he or she would interact with groups of humans. By giving devices knowledge of the users visual attention, the user can manage his or her limited attention when interacting with devices using the same techniques used to manage communications with groups of people.

AuraLamp is an example of an EyePliance [19] and can been seen in Figure 3.13. The system is a common lava lamp augmented with an Eye Contact Sensor and a microphone connected to a speech recognition engine. When the user looks at the lamp, the eye contact sensor indicates that the lamp is receiving visual attention and should subsequently respond to voice commands. Because the user's visual attention is used to indicate which device should receive the given speech command, a simple speech lexicon (simply "on" or "off") can be used to control the lamp even when multiple EyePliances are in close proximity to each other.

Interference Problems of Multiple EyePliances

The Eye Contact Sensor embedded in an EyePliance consists of a camera and an IR light source [30]. When two EyePliances are placed within 80 degrees of visual angle from one another, the computer vision algorithm of either EyePliance may not be able to conclude which EyePliance the user looks at. This is because light sources interfere with one another. When the user is looking at EyePliance A, the glint produced by

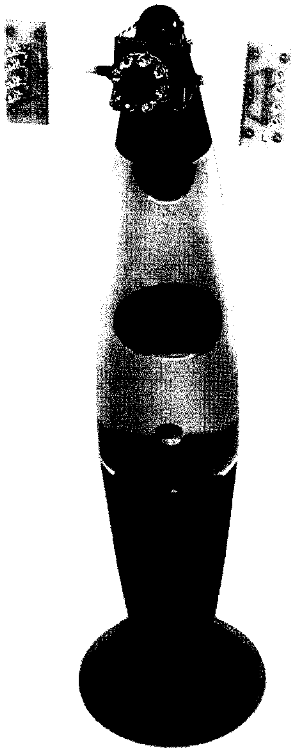

Figure 3.13: AuraLamp is an EyePliance that responds to speech commands only when the user is looking at it [19].

EyePliance A may in fact appear close to the pupil center, as seen from EyePliance ance Bs perspective [30].

Chapter 4

ViewPointer: A Simpler Approach

ViewPointer is a new head-mounted eye contact sensor that can be used in ubiquitous computing environments for a fraction of the price of previous systems. ViewPointer can be thought of as a modification of the Eye Contact Sensor, where the camera is moved from the object to an eyepiece worn on the user's head (see Figure 4.1). This simplifies the computer vision algorithm necessary for tracking the user's pupil and corneal reflections, removing the need for the on-axis, off-axis flashing used by the eye contact sensor. Any real-world object can be augmented with eye contact sensing simply by embedding a small infrared (IR) tag. We also developed a novel encoding technique that is used to uniquely identify each tag.

Like the Eye Contact Sensor, ViewPointer presents an inexpensive, calibration-free approach to eye contact detection. To detect eye contact, ViewPointer considers whether the reflection of an IR tag on the cornea appears central to the pupil. When it does, the user is looking at the tag. Our research shows this method is robust across users and camera angles at up to 80 degrees from the visual axis of the eye.

4.1 A Calibration-Free Wearable Eye Tracker

ViewPointers wearable eye tracking camera design, shown in Figure 4.1, was based on an off-the-shelf USB 2.0 snake camera, mounted on an off-the-shelf bluetooth microphone headset. The headset attaches to the users ear and has a short flexible boom which extends beyond the users eye. The booms digital camera is fitted with an IR filter, and pointed towards the users eye. The microphone and speaker headset allow for wireless speech recognition and communications with a computer. Our current prototype camera connects via USB to a 16.7 x 10.8 x 2.6 cm Sony U70 PC that weighs 0.5 kg and is carried in the users pocket to provide computer vision and other services. When fitting the device, little configuration is required as the system requires no calibration, and places no special constraints on positioning of the camera. The only requirement is that the camera has a clear line of sight with the users pupil, typically within a 45 degree angle of the users head orientation (see Fig 4). When using the device, the camera may jostle due to normal head movements. This does not affect the performance of the eye tracker.

4.2 Tracking the Eye

Because the camera is mounted close to the users eye, there is no need for background subtraction with View- Pointer. Therefore, the bright/dark pupil subtraction method of the PupilCam need not be deployed. Instead, inexpensive thresholding techniques can be used to extract the dark pupil and IR tag reflections from the image of the eye. This has the added benefit that it allows both the temporal and spatial resolution of the camera to be preserved, rather than cut in half by the alternating use of on-axis

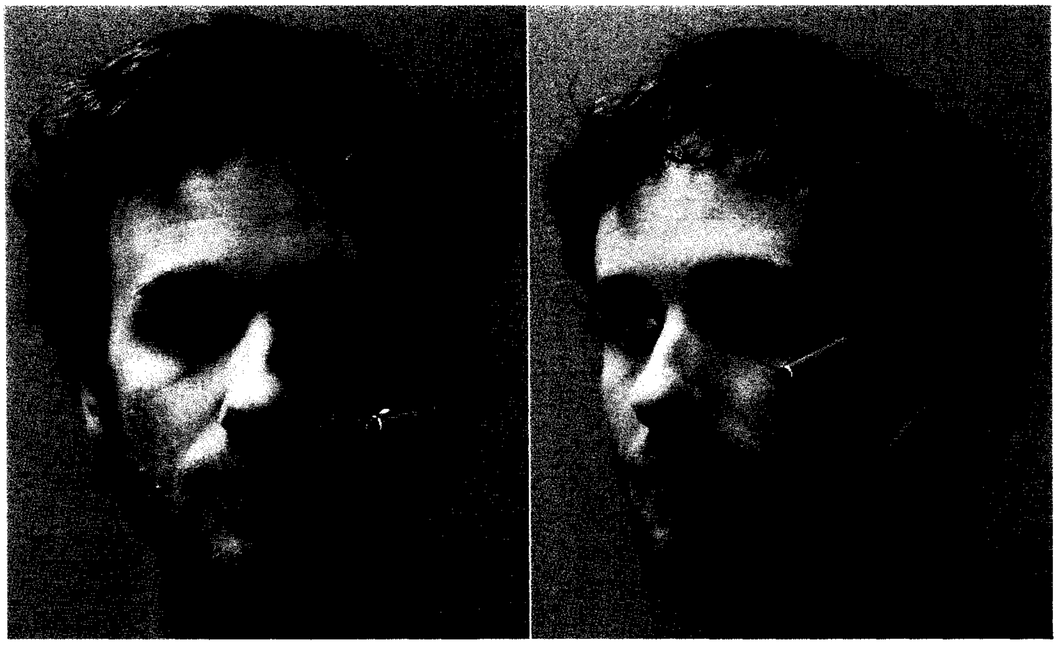

Figure 4.1: The ViewPointer headset.

and off-axis LEDs. This allowed us to design an encoding algorithm used to uniquely identify tags, which is discussed later in this section. In addition, since the camera is close to the eye, the algorithm works well even at low resolutions. Currently, the system is configured to run at a resolution of 640 x 480 pixels.

4.3 Detecting Eye Contact

Figure 4.2 shows the relationship between the pupil and the corneal reflection of a single IR LED, as observed by the camera. Even at a large camera angle, the reflection appears in the center of the pupil when the user is looking directly at the LED. This is because the pupil is set back slightly from the cornea. The cornea functions as a lens that bends the image of the pupil towards the incoming ray from the tag.

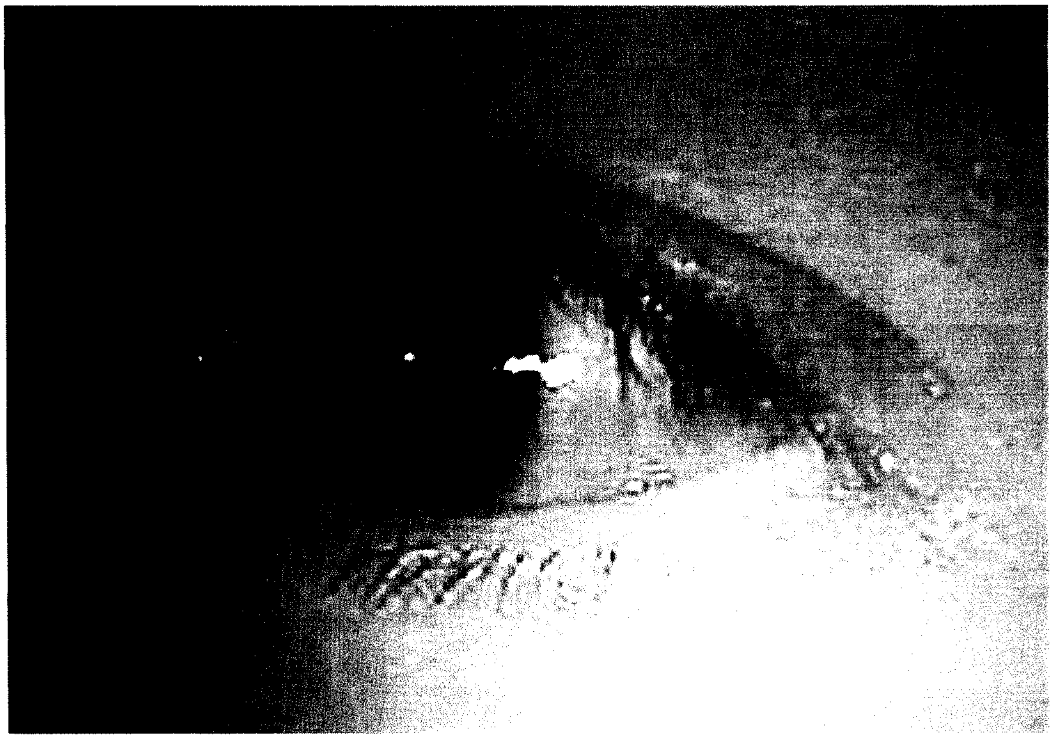

Figure 4.2: Image from the head mounted camera monitoring the users eye. The reflection of a tag can be seen in the center of the users pupil, indicating the user is looking at a tagged object.

This phenomenon allows humans to obtain a field of view of almost 180 degrees. Eye contact is reported when the distance between the reflection of the tag and the center of the pupil is less than a specified threshold. This threshold can be adjusted to alter the sensitivity of the detection algorithm. Additionally, the system is insensitive to movement by the tagged object. As long as the user tracks the object with his or her eyes, the reflection of the tag will stay in the center of the pupil. Moreover, any other tags will appear to be moving across the cornea, making the task of tracking a moving object much easier than with a calibrated eye tracker.

4.4 Tags

Figure 4.3 shows a typical IR tag, 1.5 cm in size and height, for mounting on an object that is tracked with the View- Pointer system. A tag consists of two small LEDs with a 3V cell battery and circuit. The LEDs do not emit any visible light, which makes them easy to conceal in objects, as long as direct line of sight is maintained. The small circuit allows tags to pulse according to a binary code (e.g., 101) that allows the tag to be uniquely identified by the ViewPointer system. Each cycle of the modulated binary code is distinguished by a separator code that consists of a series of zeros of the same length, with one bit padding on either end. For example, with a three bit code, this separator would consist of 10001. The Nyquist theorem [24] maintains that a signal must be sampled at double its transmission rate. Because the algorithm used to extract the pupil and tag reflections from the image of the eye is inexpensive, the frame rate of the camera is the determining factor for the rate at which data can be transmitted. The current implementation has a frame rate of 28 frames per second. Therefore, data can be transmitted at a rate of at most 14 bits per second. It is assumed that both the transmitter and receiver have knowledge of both the transmitters bit rate and tag length.

One of the drawbacks of this encoding technique is that a tradeoff exists between the number of unique tags and the time a user must look at a tag. The current configuration operates at a frame rate of 28 frames per second, yielding a transmission rate of 14 bps. If an application requires 8 unique tags, that means a tag code must be 3 bits long, with a separator of 5 bits. Given our bandwidth restrictions, the user must fixate on the tagged object for 570 ms before its code can be identified. If an application requires 64 unique tags, each tag code must be 6 bits in length, with an

Figure 4.3: A ViewPointer tag compared to a US penny.

8-bit separator. In this case, the user must fixate on the tag for a minimum of 1 second before its code can be identified. However, these times are well within the range of normal human fixations, which are typically between 100 msec and 1 second [34].

4.5 Transmitting More Complicated Data Strings

The data transmitted by the tags is not restricted to only unique identifiers. In fact, any binary data can be encoded in a tag, including URLs, text, multimedia, etc. However, in most cases the data stream will be substantially larger than that of the unique identifier. Given our current bandwidth limitations, we chose to increase the transmission speed by applying a parallel encoding scheme. We space multiplexed data transmission by mounting 5 tags, separated by about 6 degrees of visual arc, in a star formation (see Fig 7). To transmit a URL, ASCII characters are coded into a 6-bit binary number, with each code corresponding to the letters sequence in the roman alphabet. Our coding scheme also supports common special characters and digits. Subsequently, a URL is separated into chunks by dividing it by the number of tags. The system assumes all URLs are of type http://. The URL www.chi2005.org would thus be split into the following five strings (www, .ch, i20, 05. and org). The first tag sequentially beams the characters in the first string, the second tag the characters of the second string, etc. Each tag loops its string of 3 characters indefinitely, with a binary null to indicate the start of a new cycle. Including an 8-bit separator, this yields a string size of 4 14-bit numbers, or 56 bits per tag. With a bandwidth of 14 bps, the overall time needed to transmit this data is reduced to four seconds for the entire URL. Bandwidth is further increased by assuming www. and .com when no dots are present in the URL.

This method gives us a means of providing functionality similar to that of RFID tags to the user, with the chief distinction that recognition is directional. Moreover, detection is based on the actual interest of a user in the associated information, as it is correlated with his or her reading behavior. For example, this allows URLs that are printed onto a surface to be automatically stored. Additionally, functionality at a URL, such as java applets, can be downloaded and executed upon a fixation of the eye.

4.6 Detection Accuracy

Initial evaluations of the system suggest that standard dual- LED tags can be detected from a distance of up to 3 m. At 1 m distance, tags must be at least 10 cm apart.

If tags are too close together they will blend into a single corneal reflection, causing the encoding scheme to fail. A significant drawback of our current implementation is that glares prevent the camera from getting a clear line of sight with the users pupil if the user is wearing glasses. However, contact lenses do not appear to affect the systems performance. The system is tolerant to head movement in any direction, as long as the user retains a fixation within approximately 6 degrees from the tag. This requirement is inherent to the system since reflections of tags within 6 degree of each other tend to blend together on the user's pupil. It is also tolerant to substantial movement and repositioning or change in the angle of the headset camera, as long as the camera retains vision of the pupil and stays within a 45 degree angle from the visual axis of the eye.

Chapter 5

Implementation

ViewPointer is implemented in three main components: the hardware, a novel framework used to design and construct corneal reflection eye trackers, and the eye contact sensing system built upon that framework.

5.1 Hardware

The entire ViewPointer headset can be seen in Figure 4.1. The ViewPointer headset consists of a lightweight USB snake camera that is attached to a wireless Bluetooth microphone and earpiece. The "snake" portion of the snake camera serves as a boom, which extends beyond the user's eye. The camera has a frame rate of 28 frames / sec at a resolution of 640 x 480 pixels. The camera was modified to include an IR filter on the lens and also a small IR led was added to provide sufficient ambient light such that the pupil could be seen. A USB cable runs from the camera to a computer, which powers the camera and the LED and also handles the computer vision. The microphone and earpiece use the standard Bluetooth Headset profile to communicate with the computer, and are powered with an integrated battery. The system was developed and tested on a laptop with a 1.5 GHz Pentium M running Windows XP Professional. The laptop is equipped with a WiFi 802.11g connection, which enables network connectivity.

To put on the headset, the user simply attaches the microphone and earpiece to the ear. The system is lightweight and fits comfortably without much movement during normal use. To position the camera, the user bends the boom such that the camera has a clear line of sight with the eye, but is comfortably out of foveal view.

Later versions of the system could be wireless and connect to a cell phone or PDA carried in the user's pocket. This configuration would allow use in mobile scenarios where connection to a laptop is not desirable. In addition, cell phone connectivity can provide internet access.

ViewPointer tags are inexpensive IR LEDs, which are currently connected to a computer through a "Phidget" interface [7]. This allows the blinking of the tag to be controlled programmatically by the computer. Later versions will be implemented totally with onboard hardware without any connected computer.

5.2 An Eye Tracking Framework

ViewPointer is built on top of a novel framework that can be used for a variety of eye tracking purposes, where ViewPointer is provided as a reference implementation. The framework provides an event-driven API with "Listener" interfaces that notify implementing classes of the location of the pupil and corneal reflections after every frame, as well as higher-level concepts such as eye contact with tags, URLs, etc. Also, polymorphic integration with various corneal reflection-based hardware configurations

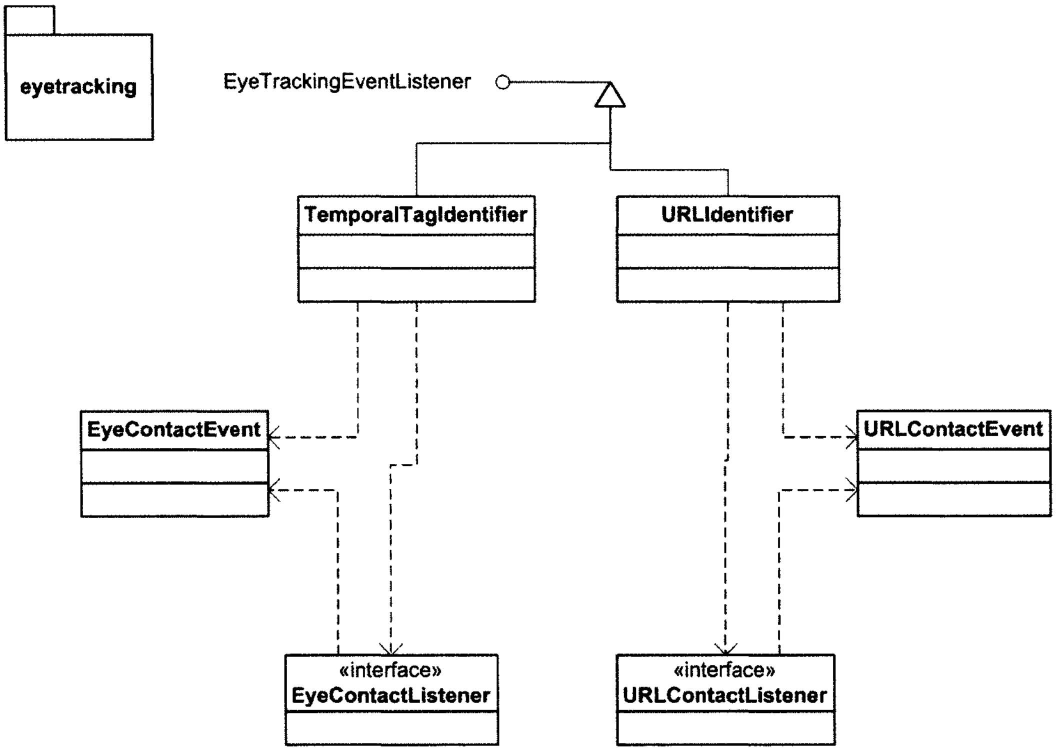

Figure 5.1: A UML class diagram of the root package in the ViewPointer framework.

is possible. In this way, the framework can be thought of as the "glue" that binds the hardware to a tracking algorithm, and then to an application, but does so in a way that does not force any specific implementation at any of the levels. The framework is implemented with Java Standard Edition version 5.0. UML class diagrams illustrating this framework can be seen in Figures 5.1 and 5.2.

The EyeTracker (Figure 5.2) abstract class is used to provide integration with low-level computer vision components. Classes that extend the EyeTracker class must report EyeTrackingEvents, which correspond to a single frame from a camera. These events consist of pupils, corneal reflections, and a timestamp. Pupils and corneal reflections are provided as extensions of Ellipse2D objects from the "java.awt.geom"

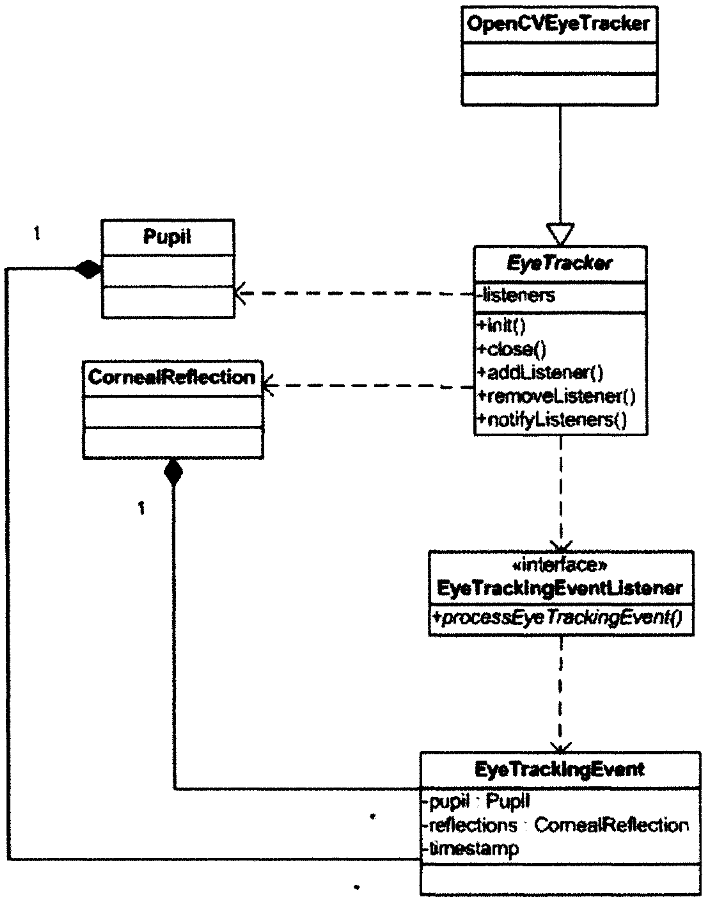

Figure 5.2: A UML class diagram of the eye tracking package in the ViewPointer framework.

package, which provides an existing math library and allows easy integration with existing graphics and windowing packages such as Java2D, AWT, Swing, SWT, etc. Applications receive these notifications by implementing the "EyeTrackingEventListener" interface and registering with an EyeTracker (Figure 5.2).

Typically, classes that correlate pupil and corneal reflections to eye movements receive EyeTrackingEvents. These algorithms then report higher-level events such as eye contact with tags, eye contact with URLs, gaze points correlated with screen coordinates, etc. Standardized reporting of eye contact with tags and URLs are provided through the "EyeContactListener" and "URLContactListener" interfaces respectively (Figure 5.1).

5.3 ViewPointer Implementation

5.3.1 System Architecture

One of our goals when designing the software architecture for ViewPointer was run-time modularity. We wanted to isolate the computer vision task of detecting the pupil and corneal reflection because this low-level operation could potentially be integrated into an onboard hardware processing component at a later time. In addition, we plan to use the pupil and corneal reflection tracking technique for future projects. Therefore, we used a client-server architecture when implementing ViewPointer, where the computer vision component is implemented in the server and the higher-level eye contact detection and reporting is performed by the client. Together, these two components compose an implementation of the EyeTracker abstract class (Figure 5.2).

5.3.2 Server

The ViewPointer Server handles the low-level image processing necessary to extract the pupil and corneal reflections from the image collected from the camera. The pupil and corneal reflections are then streamed out to the client as ellipses over a specified port using WinSock. The image capture and processing algorithm is implemented with the Intel OpenCV Computer Vision Library version 3.1 [10]. The server was developed with Microsoft Visual C++ Version 6.0 and has been tested on machines running Microsoft Windows XP.

Detecting the Pupil and Corneal Reflection

The pupil and corneal reflections are found using simple adaptive thresholding techniques. To find the pupil, the darkest region of the image that is roughly the expected size of the pupil is determined. The image histogram is used to locate this region, and can be seen in Figure 5.3. The hill on the left side of the histogram is the pupil. A sliding window is used to find the valley which separates this hill from the rest of the histogram and sets a threshold at the value found at the minimum of this valley. A call to the "cvFindContours" function in OpenCV then provides a sequence of all the objects in the scene that are darker than the threshold. The region that most appropriately fits the expected size of the pupil is chosen.

To capture the corneal reflections, a region of interest is set for the pupil. The histogram of this region of interest can be seen in Figure 5.4. The small peak on the far right of the histogram represents a tag. A sliding window is used to identify this peak and sets the threshold shortly before it. A call to the "cvFindContours" function in OpenCV then provides a sequence of all the objects in the pupil that are

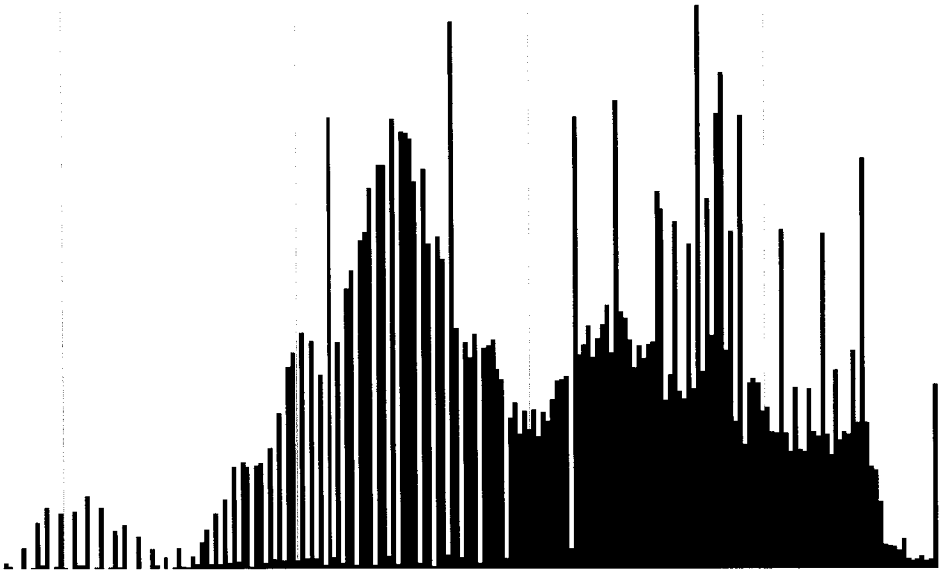

Figure 5.3: The intensity histogram of the image of the user's eye as seen from the camera on the ViewPointer headset.

lighter than the threshold. All of these objects are reported as corneal reflections.

Protocol

The server multi-casts the pupil and corneal reflections in terms of their bounding rectangle. Coordinates are provided in ASCII format and in pixel coordinates from the camera frame. At the start of each frame, the pupil is sent in the format:

PX,Y,W,H[RET]

Each corneal reflection is then provided in the format:

CX,Y,W,H[RET]

The end of a frame is indicated with the following:

#[RET]

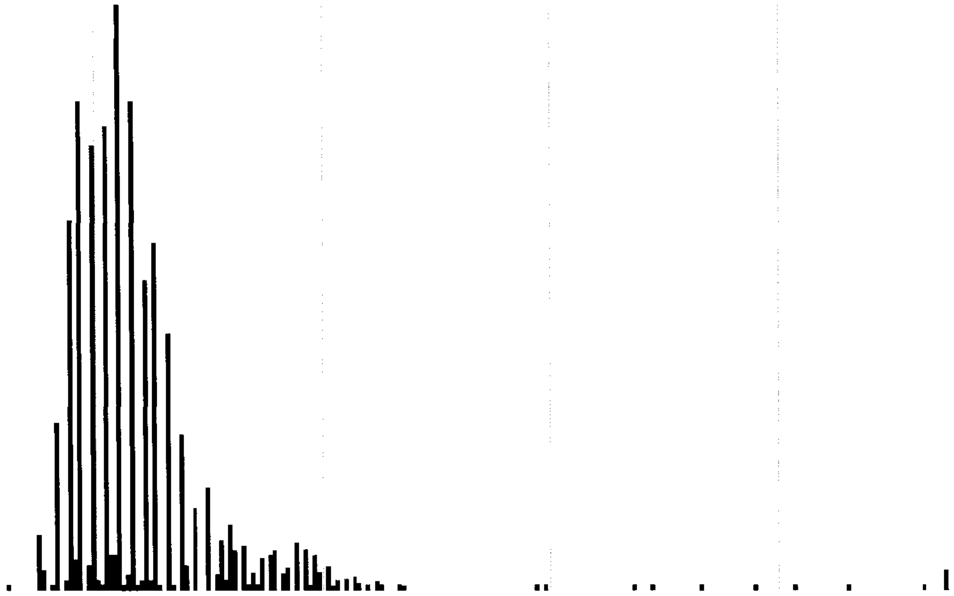

Figure 5.4: The intensity histogram of the image of the user's pupil when looking at a tag.

5.3.3 Client

The client provides integration with the eye tracking framework and references the server over a network. The client is built with Java Standard Edition Version 5.0. The OpenCVEyeTracker class is used to communicate with the server and extends the EyeTracker abstract class (Figure 5.2). The OpenCVEyeTracker class uses the "java.net" and "javax.net" packages for network communication with the server. The OpenCVEyeTracker converts each frame consumed through the protocol described in the previous section into a new EyeTrackingEvent and immediately notifies its registered listeners. Therefore, the OpenCVEyeTracker generates events at roughly the same rate as the camera frame rate.

5.3.4 Tag Identifier

The TemporalTagIdentifier class (Figure 5.1) implements the EyeTrackingEventListener and identifies when the user is looking at a tag described in Section 4.4. The TemporalTagIdentifier must be initialized with the rate at which the tags are blinking, and also the threshold of the distance between the pupil center and a corneal reflection which determines when eye contact is made. Because the pupil and corneal reflections are reported as extensions to the Ellipse2D class, the corneal reflections can be compared to the center of the pupil with the math library afforded by the "java.awt.geom" package. A "sliding window" is used to detect tags within the sequence of EyeTrackingEvents.

The TemporalTagIdentifier also keeps a Collection of "EyeContactListeners" that respond to EyeContactEvents. When a tag is identified, an EyeContactEvent is raised that indicates the pattern of the tag.

5.3.5 URL Identifier

The URLIdentifier class reports eye contact with URLs encoded with the technique described in Section 4.5. The same "sliding window" technique used to identify tags is used, however the center of the pupil is compared to the centroid of the array of reflections used to multiplex the URL. The URLIdentifier maintains a Collection of "URLContactListeners" that respond to URLContactEvents, which are raised when the user makes eye contact with a URL. URLs are reported as instances of the java.net.URL class.

Chapter 6

Ubiquitous EyePliances

ViewPointer has a number of benefits over traditional Eye Contact Sensing. First, it allows any object to become an eyepliance. Second, it allows identification of users, and third, it allows transmission of data to the user.

6.1 Cheap Augmentation of Any Object

While Eye Contact Sensors offer a major reduction in cost over previous eye trackers, an environment with n EyePliances will require n Eye Contact Sensors. Because each Eye Contact Sensor contains a high resolution camera and must connect to computing resources that handle computer vision, this is a costly solution for applications with many appliances. ViewPointer addresses this problem by offloading the camera, as well as the associated computer vision, to the person, rather than the EyePliance. ViewPointer contains only as many cameras and computing resources as there are users in the environment. Each EyePliance only requires an inexpensive IR tag composed of two infrared LEDs, a circuit and battery. Therefore, it allows us to turn any regular object or person into an EyePliance by adding only a small, unobtrusive tag.

6.2 Easy Identification

Another benefit of ViewPointer is that it allows easy detection of the onlooker. While Eye Contact Sensors can detect when a user is looking at an EyePliance, they do not know who is making eye contact. This leads to problems in the case of multi-user scenarios. For example, a Look-To-Talk EyePliance such as AuraLamp [19] will misinterpret eye contact by user A as meaning it should listen to spoken commands originating from user B. EyePliances that use ViewPointer can also more readily track multiple users in environments containing multiple EyePliances because they are personalized. ViewPointer allows any speech recognition engine to be carried by the user, rather than the EyePliance. This allows superior handling of personalized acoustic models for speech recognition, and reduces the amount of ambient noise picked up by the microphone. Similarly, the vocabulary or language used in the speech recognition system can be customized to fit each specific user. For example, one user may wish to address an EyePliance in English while another may wish to speak Japanese. People can be turned into EyePliances by mounting an IR ID tag onto their ViewPointer or clothes. This allows other ViewPointer systems to identify not only when their user is looking at another person, but also to uniquely identify that person. As such, a ViewPointer system provides functionality similar to ECSGlasses, and can detect ad-hoc social networks by tracking mutual eye contact patterns between users. Unlike ECSGlasses, eye contact detection is mutual, and does not necessarily interfere with EyePliance operation.

6.3 Directional Private Transmission of Information

With traditional EyePliances, the camera is mounted on the EyePliance rather than the user. Therefore EyePliances are not capable of broadcasting digital information. Although Shell et al. [17] discusses the use of RFID tags for identifying users, there are obvious downsides to this method. RFID tags are not directional, and not attentive. They transmit information whenever a reader is in close proximity. This means a user could potentially pick up information that is irrelevant to his or her task situation. The use of RFID tags for identifying users carries with it a privacy risk, in that other readers can easily pick up on any information transferred to the user. Similar problems exist with traditional EyePliances, which may be seen to encourage ubiquitous surveillance through dispersement of cameras in the environment. By contrast, ViewPointer allows any object to transmit data, but only upon being looked at. Information picked up by ViewPointer is completely private, as the receiver is worn by the user. Other systems cannot read the cornea of the user from a typical distance of use.

As such, ViewPointer allows for greatly extended interactive scenarios. A few of of these scenarios are discussed in the next chapter.

Chapter 7

Scenario: Everyday Uses

ViewPointer allows any object or person to become an EyePliance. The microphone attached to the ViewPointer headset allows a user to Look-To-Talk to any tagged object, with speech feedback being provided in the users headset. The following scenario illustrates some of the possible applications of ViewPointer, as applied to everyday eye contact sensing objects:

Ted is shopping in Manhattan. He's wearing a handsfree bluetooth headset augmented with a ViewPointer, attached to his PDA phone. The PDA phone has a wireless internet connection, and is augmented with a unique IR identifier tag. As Ted walks past the Apple store in Soho, he notices an advertisement in the storefront window for one of their latest products, a Bluetooth iPod that would connect to his wireless headset. The poster is augmented with a ViewPointer tag that broadcasts a URL to the product website.

The tags are read by Ted's ViewPointer as he examines the poster. Ted wants to buy the product, but would like to query for reviews. He looks at his PDA, and selects a Google query on the URL obtained from the poster. Google lists the product website, *but Ted instead taps a link to find webpages containing the URL. Google displays a link to an up-to-date wikipedia article on the new product, which informs him it is indeed compatible with his Bluetooth headset. Ted changes the query menu on his webbrowser from Google to MySimon.com, which finds product comparisons on the basis of the URL. He discovers that the iPod is available online for less at amazon.com. He hits the Buy Now button, pulling his credit card from his wallet, which is augmented with a tag that uniquely identifies the card to his PDA. The PDA retrieves the associated credit card number, and enters it automatically to complete the sale.*

Ted wants to find out what the shortest route is from the Apple store to the nearest subway station. He looks at the street number on the store front, which is augmented with a URL tag that provides the intersection as a query string. He looks at his PDA, selecting a google map query from the browser menu to obtain a map of the area. Ted clicks a button to reveal the subway stations on the map. The nearest one is only a block from Broadway. On the subway, Ted notices a friend who is also wearing a ViewPointer. When they make eye contact, the ViewPointers exchange unique IDs. Ted pulls out his Stowaway Universal Bluetooth Keyboard [19] and sits down opposite his friend, who does the same. As the two make eye contact, the keyboards connect to each other's PDA, causing words entered to be translated by text-to-speech and spoken in the other persons headset. This allows Ted and his friend to have a completely silent and private conversation in a crowded and noisy public space. When Ted gets home he enters his house, looks at the lights and says On. The speech recognition engine interprets his command within the context provided by the tags mounted near the lamp, sending a command to the switch through X10 [24]. While waiting for his wife to arrive, he decides to prepare dinner. As he is busy *cooking, he looks at his answering machine, which shows 3 messages. The answering machine is augmented with an ID tag that allows the speech recognition system in the headset to shift its context from the light system to the answering machine. Ted says Play, causing his answering machine to play the first message. It is Teds mother. The message is lengthy, so Ted decides to play some music. He look at the kitchen radio, also augmented with an ID tag, and says: Play. As the sound from the radio fills the room, the answering machine plays the next message. It is his wife informing him that she will not be home for dinner.*

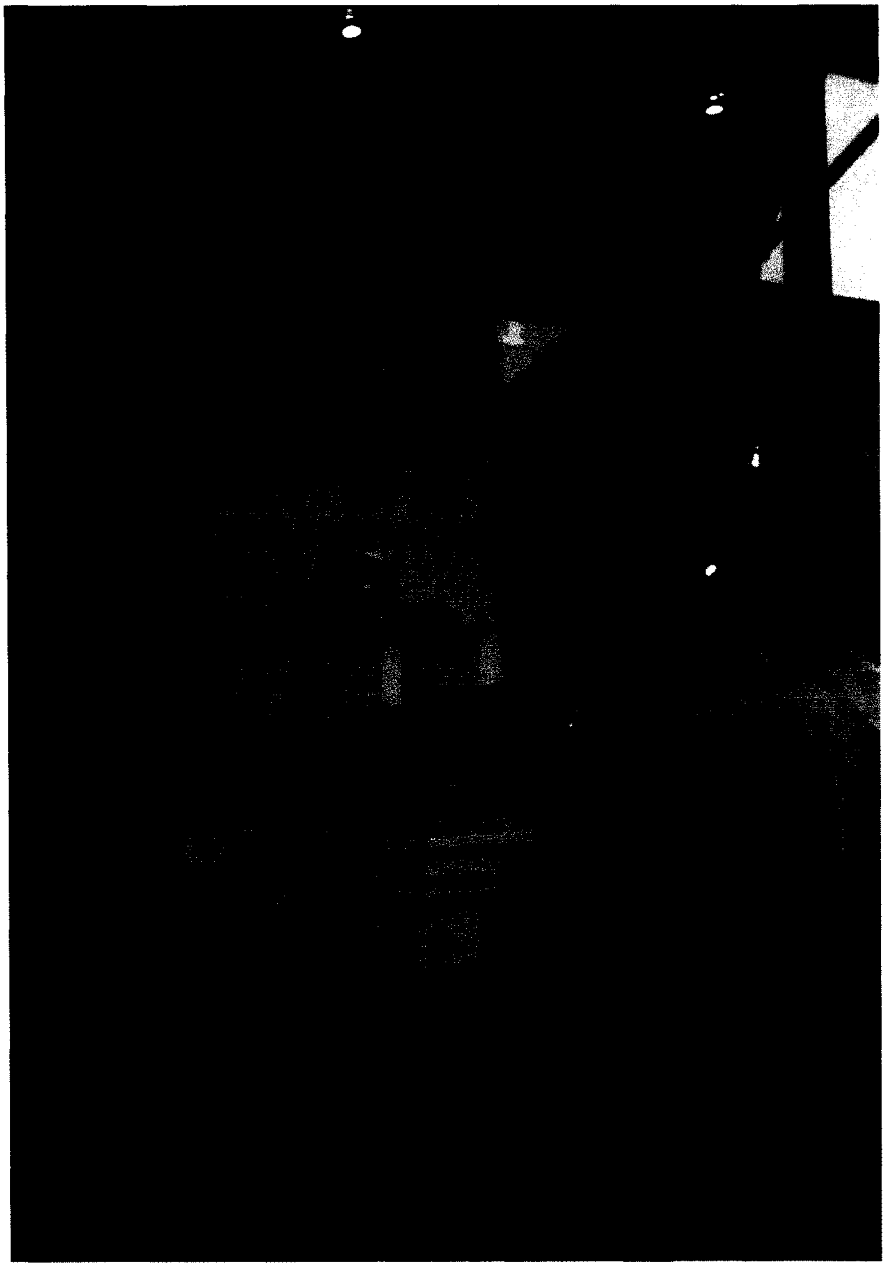

Figure 7.1: A user looking at a poster augmented with an invisible ViewPointer URL tag mounted behind the Virgin logo.

Chapter 8

Discussion

In examining the above scenario, we are particularly interested in analyzing how the eyes may provide context to action by other forms of input. The notion of providing context to action was investigated early-on by Guiard with his Kinematic Chain (KC) theory [8]. He saw the hands function as serially assembled links in a kinematic chain, with the left (or non-dominant) hand as a base link and the right (or dominant) hand as the terminal link. With regard to providing context to action, the KC model has a number of relevant properties: (1) the left (non-dominant) hand sets the frame of reference, or context, for action of the right hand. (2) the granularity of action of the left (non-dominant) hand is coarser than the right. (3) the sequence of motion is left (non-dominant) followed by right (dominant). (4) the right hand tends to be dominant because it is typically the terminal link. If we include the eyes in this model, we notice that their activity provides input to, and therefore precedes, the activities of the non-dominant hand in the chain. This provides the following number of observations. (1) Eye fixations provide one of the best available estimates of the focus of user attention. This is because they indicate the location of the window through which the user's mind interprets the task. As such, the eyes set the frame of reference for action of the other links in the kinematic chain. (2) Although the eyes are capable of positioning with great accuracy, the granularity of eye movements tends to be coarser than that of the non-dominant hand. This is because the eyes tend to jump from context to context (i.e., visual object to visual object). (3) When a task is not well-rehearsed, humans tend to look at the object of manual action before engaging the kinematic chain. The sequence of motion is eyes, then left (non-dominant), then right (dominant) hand. (4) The eyes thus provide context to action performed by the limbs that end the kinematic chain. From this model, we can derive a number of principles:

1. Principle 1. Eye contact sensing objects provide context for action. Application of the KC model implies that tagged objects best act as passive providers of information that set the context for action performed with another method of input. For example, the URL of the Apple Store poster in our scenario did not cause the PDA to immediately look up a map. The act of specifying what to look up is best left to the hands. If we take the metaphor of a GUI tool palette, the eyes would be used to select the drawing tool, not to perform the actual drawing task [9].

2. Principle 2. Design for input = output. User interface objects should provide information that naturally attracts the eyes. By doing so, the pointing action becomes secondary to the act of observing information. This reduces pointing errors, and minimizes cognitive load required to perform the pointing task. It is perhaps this principle that makes users refer to interactive eye tracking technologies as magical [40]. For example, the iPod poster in the Apple store

naturally captured the attention of the user. The primary reason for looking was to observe the visual information on the poster. The transmission of the URL was a side effect of this activity that came at no apparent cost to the user.

3. Principle 3. Avoid direct action upon eye contact. Eye trackers suffer from what is known as the Midas Touch Effect [14]. This is caused by overloading the visual input function of the eye with a motor output task. It occurs chiefly when an eye tracker is used not only for pointing, but also for clicking targets. The Midas Touch effect causes users to inadvertently select or activate any target they fixate upon. The Midas Touch effect can, in general, be avoided by issuing actions via an alternate input modality, such as a manual button or voice command. More generally, eye contact sensing objects should avoid taking direct action upon receiving a fixation. In our scenario, the kitchen light, answering machine and radio did not act upon looking. Instead, looking provided the context for the ensuing voice command.

4. Principle 4. Design for Deixis. The eyes are ill-suited for pointing at coordinates in a visual space. Rather, the use of eye input should be designed such that it corresponds to visually meaningful and discrete targets. This principle allows for the eyes to function as a means of indicating the target of commands issued through other means. Examples include the use of eye contact to direct instant messaging traffic in the subway scenario, and the use of eye contact to specify the target of speech commands in the kitchen scenario.

Chapter 9

Future Work

We are currently looking at ways to further improve ViewPointer technology and its applications in real world scenarios.

9.1 Increased Frame Rate

We are exploring ways to increase the frame rate of the camera. This will facilitate more widespread adoption of ViewPointer as it will allow our encoding algorithm to identify tags with larger bit lengths. Large bit lengths potentially allow unique encoding of any object such that it could be referenced in an online database in ways similar to RFID. An increased frame rate also reduces the amount of time the user must fixate on an object before the tag can be uniquely identified. Our implementation works reliably at 28 frames per second. With a frame rate of 100 Hz, we can transmit up to 24 bits per second (including the separator bits), allowing the system to identify tags with approximately 17 million unique IDs in a one second fixation.

9.2 Printable Tags

ViewPointer tags are unobtrusive, wireless, and very inexpensive. However, there exist objects that do not allow the incorporation of even the smallest IR tag. Placing a tag on a product during the manufacturing process is undesirable, especially for low priced items, because the addition of a tag adds marginal cost to each unit. Similarly, paper magazines and other thin materials would not benefit from the addition of our comparatively bulky tags.

To address such concerns, we are developing ways to print tags onto objects using IR reflective ink. IR reflective ink is invisible to the naked eye, cheap, and can be applied to most materials using the same printing technologies used for any other type of inks. Such tags could potentially be printed onto the packaging of every day items in the same way that UPC codes are currently printed. This could allow items to be tagged on several sides without affecting the objects appearance. Additionally, paper items such as magazines could be tagged, which offers great promise. To detect a printed tag, we would deploy techniques similar to those used for URL ViewPointer tags, with the distinction that the IR light source that illuminates the tags would be mounted on the ViewPointer headset instead. However, printed tags cannot be modulated in the same way as our active LEDs. For this reason, we are currently developing a space-multiplexed encoding technique similar to barcode.

9.3 Existing IR Sources

Another area of improvement we are exploring is the use of pre-existing IR light sources in the real world. Some examples of these include regular light bulbs as well as ambient sunlight passing through windows. We are exploring ways in which these natural sources of IR light can be harnessed to reference objects in a room, or modulated to transmit information to the user.

Chapter 10

Summary

We have presented ViewPointer, a wearable eye contact sensor that provides deixis towards everyday ubiquitous computing devices. ViewPointer consists of a small wearable camera mounted on a regular Bluetooth headset. ViewPointer allows any real-world object to be augmented with eye contact sensing capabilities simply by embedding a small infrared tag in the object. The headset camera detects when a user is looking at an infrared tag by determining whether the reflection of the tag on the cornea of the users eye appears sufficiently central to the pupil. ViewPointer also includes a wireless microphone headset that can be used to interpret voice commands. The system not only allows any object to become an eye contact sensing appliance, it also allows identification of users and transmission of data to the user by the object. By pulse code modulation of tags we can uniquely identify objects looked at, as well as transmit data objects such as URLs. We analyzed scenarios of application, providing five design principles for eye contact sensing input: (1) Eye contact sensing objects provide context to action; (2) Eye contact sensing works best when input = output; (3) Avoid direct action upon eye contact; (4) Design for deixis, rather than pointing and (5) The eyes are best used to open and close communication of content.

Bibliography

[1] *Perception and Communication.* Pergamon, Oxford, England, 1958.

[2] *Treatise on Physiological Optics (translated and edited by J. P. C. Southhall).* Dover, New York, USA, 1962.

[3] M. Argyle and M. Cook. *Gaze and Mutual Gaze.* Cambridge University Press, London, 1976.

[4] Nathan Cournia, John D. Smith, and Andrew T. Duchowski. Gaze- vs. hand-based pointing in virtual environments. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 772–773, New York, NY, USA, 2003. ACM Press.

[5] J. A. Deutsch and D Deutsch. Attention: Some theoretical considerations. *Psychological Review*, 70(1):80–90, 1963.

[6] Andrew T. Duchowski. *Eye tracking methodology.* Springer, 2003.

[7] Saul Greenberg and Chester Fitchett. Phidgets: easy development of physical interfaces through physical widgets. In *UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 209–218, New York, NY, USA, 2001. ACM Press.

[8] Y Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19(4):486–517, 1987.

[9] Anthony Hornof, Anna Cavender, and Rob Hoselton. Eyedraw: a system for drawing pictures with the eyes. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1251–1254, New York, NY, USA, 2004. ACM Press.

[10] Intel Inc. Opencv computer vision library, 2005. http://www.intel.com/research/mrl/research/opencv/.

[11] LC Technologies Inc. Eye gaze system, June 2005. http://www.eyegaze.com.

[12] Polhemus Inc. Visiontrak etl-400, June 2005. http://www.polhemus.com/VisionTrak.htm.

[13] Tobii Technology Inc. Tobii 1750, June 2005. http://www.tobii.se.

[14] Robert J. K. Jacob. What you look at is what you get: eye movement-based interaction techniques. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 11–18, New York, NY, USA, 1990. ACM Press.

[15] William James. *The Principles of Psychology.* Holt, New York, 1890. Quoted in [Grudin 90].

[16] Stephen M. Kosslyn. *Image and Brain: The Resolution of the Imagery Debate.* MIT Press, Cambridge, MA, 1994.

[17] S. C. Levinson. *Pragmatics.* Cambridge University Press, Cambridge, England, 1983.

[18] Paul P. Maglio, Teenie Matlock, Christopher S. Campbell, Shumin Zhai, and Barton A. Smith. Gaze and speech in attentive user interfaces. In *ICMI '00: Proceedings of the Third International Conference on Advances in Multimodal Interfaces*, pages 1–7, London, UK, 2000. Springer-Verlag.

[19] A. Mamuji, R. Vertegaal, J. Shell, T. Pham, and C. Sohn. Auralamp: Contextual speech recognition in an eye contact sensing light appliance. In *Extended Abstracts of Ubicomp 03*, 2003.

[20] J. Merchant, R. Morrissette, and J. L. Porterfield. Remote measurement of eye direction allowing subject motion over one cubic foot of space. *Transactions on Biomedical Engineering*, 21(4):309–317, 1974.

[21] C.H. Morimoto, D. Koon, A. Amir, and M. Flickner. Pupil detection and tracking using multiple light sources. *Image and Vision Computing*, 18:331–334, 2000.

[22] Donald A. Norman. *Some Observations on Mental Models.* Lawrence Erlbaum Associates, 1983.

[23] D. Noton and L. Stark. Eye movements and visual perception. *Scientific American*, pages 34–43, 1971.

[24] H Nyquist. Certain topics in telegraph transmission theory. *Trans. AIEE*, 47:617–644, 1928.

[25] Alice Oh, Harold Fox, Max Van Kleek, Aaron Adler, Krzysztof Gajos, Louis-Philippe Morency, and Trevor Darrell. Evaluating look-to-talk: a gaze-aware

interface in a collaborative environment. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 650–651, New York, NY, USA, 2002. ACM Press.

[26] M. I. Posner. Orienting of attention. *Quarterly Journal of Experimental Psychology*, (32):3–25, 1980.

[27] Ted Selker, Andrea Lockerd, and Jorge Martinez. Eye-r, a glasses-mounted eye motion detection interface. In *CHI '01: CHI '01 extended abstracts on Human factors in computing systems*, pages 179–180, New York, NY, USA, 2001. ACM Press.

[28] Jeffrey S. Shell, Ted Selker, and Roel Vertegaal. Interacting with groups of computers. *Commun. ACM*, 46(3):40–46, 2003.

[29] Jeffrey S. Shell, Roel Vertegaal, Daniel Cheng, Alexander W. Skaburskis, Changuk Sohn, A. James Stewart, Omar Aoudeh, and Connor Dickie. Ecsglasses and eyepliances: using attention to open sociable windows of interaction. In *ETRA'2004: Proceedings of the Eye tracking research & applications symposium on Eye tracking research & applications*, pages 93–100, New York, NY, USA, 2004. ACM Press.

[30] Jeffrey S. Shell, Roel Vertegaal, and Alexander W. Skaburskis. Eyepliances: attention-seeking devices that respond to visual attention. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 770–771, New York, NY, USA, 2003. ACM Press.

[31] Vildan Tanriverdi and Robert J. K. Jacob. Interacting with eye movements in virtual environments. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272, New York, NY, USA, 2000. ACM Press.

[32] Akira Tomono, Muneo Iida, and Kozunori Ohmura. Method of detecting eye fixation using image processing. U.S. Patent: 5,818,954, 1991. ATR Communication Systems Research Laboratories.

[33] A.M. Treisman and G. Gelade. A feature integration theory of attention. *Cognitive Pscychology*, 12(1):97–136, 1980.

[34] Boris M. Velichkovsky and John Paulin Hansen. New technological windows into mind: there is more in eyes and brains for human-computer interaction. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 496–503, New York, NY, USA, 1996. ACM Press.

[35] R. Vertegaal, G. Van der Veer, and H Vons. Effects of gaze on multiparty mediated communication. In *Proceedings of Graphics Interface 2000*, pages 95–102, 2000.

[36] Roel Vertegaal. The gaze groupware system: mediating joint attention in multiparty communication and collaboration. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 294–301, New York, NY, USA, 1999. ACM Press.

[37] M. Weiser. The computer for the 21st century. *Scientific American*, 265(3):94–104, 1991.

[38] A.L. Yarbus. *Eye Movements and Vision.* Plenum Press, 1967.

[39] Chen Yu and Dana H. Ballard. A multimodal learning interface for grounding spoken language in sensory perceptions. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*, pages 164–171, New York, NY, USA, 2003. ACM Press.

[40] Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (magic) pointing. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 246–253, New York, NY, USA, 1999. ACM Press.