Title	Adaptive Display of Virtual Content for Improving Usability and Safety in Mixed and Augmented Reality
Author(s)	Orlosky, Jason
Citation	大阪大学, 2016, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/55854
rights	
Note	

Adaptive Display of Virtual Content for Improving Usability and Safety in Mixed and Augmented Reality

January 2016

Jason Edward ORLOSKY

Adaptive Display of Virtual Content for Improving Usability and Safety in Mixed and Augmented Reality

Submitted to the
Graduate School of Information Science and Technology
Osaka University

January 2016

Jason Edward ORLOSKY

Thesis Committee

Prof. Haruo Takemura (Osaka University)

Prof. Takao Onoye (Osaka University)

Assoc. Prof. Yuichi Itoh (Osaka University)

Assoc. Prof. Kiyoshi Kiyokawa (Osaka University)

List of Publications

Journals

1) Orlosky, J., Toyama, T., Kiyokawa, K., and Sonntag, D. ModulAR: Eye-controlled Vision Augmentations for Head Mounted Displays. In *IEEE Transactions on Visualization and Computer Graphics (Proc. ISMAR),* Vol. 21, No. 11. pp. 1259–1268, 2015. **(Section 5.3)**

2) Orlosky, J., Toyama, T., Sonntag, D., and Kiyokawa, K. The Role of Focus in Advanced Visual Interfaces. In *KI-Künstliche Intelligenz*. pp. 1–10, 2015. **(Chapter 4)**

3) Orlosky, J., Shigeno, T. Kiyokawa, K. and Takemura, H. Text Input Evaluation with a Torso-mounted QWERTY Keyboard in Wearable Computing. In *Transaction of the Virtual Reality Society of Japan.* Vol.19, No. 2, pp. 117–120, 2014. **(Section 6.3)**

4) Kishishita, N., Orlosky, J., Kiyokawa, K., Mashita, T., and Takemura, H. Investigation on the Peripheral Visual Field for Information Display with Wide-view See-through HMDs. In *Transaction of the Virtual Reality Society of Japan.* Vol. 19, No. 2, pp.121–130, 2014.

5) Orlosky, J., Kiyokawa, K., and Takemura, H. Managing Mobile Text in Head Mounted Displays: Studies on Visual Preference and Text Placement. In the *Mobile Computing and Communications Review*, Vol. 18, No. 2, pp. 20–31, 2014. **(Section 3.4)**

Peer Reviewed Conferences

1) Orlosky, J., Toyama, T., Kiyokawa, K., and Sonntag, D. ModulAR: Eye-controlled Vision Augmentations for Head Mounted Displays. In P*roceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (same as journal #1), 2015. **(Chapter 5.3)**

2) Orlosky, J., Toyama, T., Kiyokawa, K., and Sonntag, D. Halo Content: Context-aware Viewspace Management for Non-invasive Augmented Reality. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI),* pp. 369–373. 2015. **(Section 3.5)**

3) Toyama, T., Orlosky, J., Sonntag, D., and Kiyokawa, K. Attention Engagement and Cognitive State Analysis for Augmented Reality Text Display Functions. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI),* pp. 322–332. 2015.

4) Kishishita, N., Kiyokawa, K., Orlosky, J., Mashita, T., Takemura, H., and Kruijff, E. Analysing the effects of a wide field of view augmented reality display on search performance in divided attention tasks. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR),* pp. 177–186, 2015.

5) Orlosky, J., Wu, Q., Kiyokawa, K., Takemura, H., and Nitschke, C. Fisheye vision: peripheral spatial compression for improved field of view in head mounted displays. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction (SUI),* pp. 54–61, 2015. **(Section 5.2)**

6) Toyama, T., Orlosky, J., Sonntag, D., and Kiyokawa, K. A natural interface for multi-focal plane head mounted displays using 3D gaze. In *Proceedings of The Working Conference on Advanced Visual Interfaces (AVI),* pp. 25–32, 2014.

7) Orlosky, J., Kiyokawa, K., and Takemura, H. Dynamic Text Management for See-through Wearable and Heads-up Display Systems. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI),* pp. 363–370, 2013. **(Section 3.4)** **Best Paper**

Peer Reviewed Posters, Demos, Consortia, and Workshops

1) Orlosky, J., Weber, M., Gu. Y., Sonntag, D., and Sosnovsky, S. An Interactive Pedestrian Environment Simulator for Cognitive Monitoring and Evaluation. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI) Companion*, pp. 57–60, 2015. **(Section 6.5)**

2) Orlosky, J., Depth based interaction and field of view manipulation for augmented reality. In *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST),* pp. 5–8, 2014. **(Chapter 5)**

3) Orlosky, J., Toyama, T., Sonntag, D., Sarkany, A., and Lorincz, A. On-body multi-input indoor localization for dynamic emergency scenarios: fusion of magnetic tracking and optical character recognition with mixed-reality display. In *Proceedings of the 2014 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops),* pp. 320–325, 2014. **(Section 6.2)**

4) Orlosky, J., Kiyokawa, K., and Takemura, H. Towards intelligent view management: A study of manual text placement tendencies in mobile environments using video see-through displays. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR),* pp. 281–282, 2013. **(Section 3.4.9)**

5) Orlosky, J., Kiyokawa, K., and Takemura, H. Management and Manipulation of Text in Dynamic Mixed Reality Workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR),* pp. 1–4, 2013. **(Chapter 3)**

6) Kishishita, N., Orlosky, J., Mashita, T., Kiyokawa, K., and Takemura, H. Poster: Investigation on the peripheral visual field for information display with real and virtual wide field-of-view see-through HMDs. In *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI),* pp. 143–144, 2013.

7) Walker, B., Godfrey, M., Orlosky, J., Bruce, C., and Sanford, J. Aquarium Sonification: Soundscapes for Accessible Dynamic Informal Learning Environments. In *Proceedings of The 12th International Conference on Auditory Display (ICAD),* pp. 238–241, 2006.

Other Non-peer Reviewed Work

1) Orlosky, J., Adaptive Display for Augmented Reality. *The 9th Young Researcher's Retreat*, 2015. **Best Poster**

2) Orlosky, J., Toyama, T., Sonntag, D., and Kiyokawa, K. Using Eye-Gaze and Visualization to Augment Memory. In *Distributed, Ambient, and Pervasive Interactions*, pp. 282–291. 2014. **(Section 6.4)**

3) Voros, G., Miksztai Rethey, B., Vero, A., Orlosky, J., Toyoma, T., Sonntag, D., and Lorincz. A., Mobile AAC Solutions using Gaze Tracking and Optical Character Recognition. In *Proceedings of the 16th Biennial Conference of the International Society for Augmentative and Alternative Communication (ISAAC)*, 2014.

4) Vero, A., B. Miksztai Rethey, B., Pinter, B., Voros, G., Orlosky, J., Toyoma, T., Sonntag,

and D., Lorincz. A. Gaze Tracking and Language Model for Flexible Augmentative and Alternative Communication in Practical Scenarios. In *Proceedings of the 16th Biennial Conference of the International Society for Augmentative and Alternative Communication (ISAAC)*, 2014.

5) Orlosky, J., Kiyokawa, K., and Takemura, H. Automated Text Management for Wearable and See-through Display Systems. In *Proceedings of the 6th Korea-Japan Workshop on Mixed Reality (KJMR),* 2013. **(Section 3.4)**

6) Orlosky, J., Kiyokawa, K., and Takemura, H. Scene Analysis for Improving Visibility in Wearable Displays. In *Proceedings of the 16th Meeting on Image Recognition and Understanding (MIRU),* 2013. **(Section 3.4)**

7) Orlosky, J. Katzakis, N. Kiyokawa, K. and Takemura, H. Torso Keyboard: A Wearable Text Entry Device That Can Be Used While Sitting, Standing or Walking. In *Proceedings of the 10th Asia Pacific Conference on Human Computer Interaction (APCHI)*, pp. 781–782. 2012. **(Section 6.3)**

Abstract

In mobile augmented reality, a number of barriers still exist that make head worn devices unsafe and difficult to use. One of these problems is the display of content in or around the user’s field of view, which can result in occlusion of physical objects, distractions, interference with conversations, and a limited view of the user’s natural environment.

This thesis proposes the use of dynamic content display and field of view manipulation techniques as a step towards overcoming these safety and usability issues. More specifically, I introduce novel strategies for dynamic content movement, gaze depth tracking techniques for automated content management, and hands-free spatial manipulation of the user’s field of view. In combination with a number of new head mounted display prototypes, these methods can decrease the invasiveness of and increase the usability of head worn displays and related mixed and augmented reality applications. In addition to proposing frameworks and strategies for improving usability and safety, new information about the human eye, brain, and perception of virtual content are revealed and discussed.

In order to conduct an initial comparison of standard mobile interfaces to head mounted displays, I first describe pilot experiments that study user tendencies related to viewing and placing text in mobile environments. The experiments studied smartphone and head mounted display use, and tested general environmental awareness and performance between the two devices for concentration intensive mobile tasks. Results showed that head mounted displays already have some advantages in terms of environmental awareness, but more importantly, users would prefer text that is affixed to visible locations in the background rather than affixed to a single point on the head mounted display screen. Since users’ environments are constantly and dynamically changing, variables like lighting conditions, human or vehicular obstructions in users’ paths, and scene variation interfered with viewing content.

Consequently, I proposed the use of a new dynamic text management system that actively manages the appearance and movement of text in a user's field of view. Research to date lacked a method to migrate user-centric content such as e-mail or text messages throughout a user's environment while mobile. Unlike most current annotation and view management systems, my strategy utilizes camera tracking to find dark, uniform regions along the route on which a user is travelling in real time. I then implement methodology to move text from one viable location to the next to maximize readability. Because interpersonal interactions such as conversations and gestures are of particular importance, I also integrate face detection into the movement strategy to help prevent virtual content from occluding or interfering with a user’s conversation space. A pilot experiment with 19 participants showed that the text placement of the dynamic text algorithm is preferred to text in fixed location configurations. A second experiment on real time videos comparing automatic and manual text placement showed that this strategy can mimic human placement tendencies with approximately 70% accuracy. A last experiment testing the conversation management algorithm showed a 54.8% reduction in the

number of times content was evaluated as invasive when compared to a fixed layout placement.

Even though this automated movement improved the visibility and readability of content, users still often needed to switch gaze from virtual content to the real world in order to clearly view hazards or obstructions. Therefore, a fast, natural way to quickly close or dim content was necessary to prevent interference with real world activities. To deal with this need, I then proposed a novel focal-plane based interaction approach with several advantages over traditional methods. Interaction with static displays via eye tracking had often been proposed in the past, but mobile head worn devices had yet to be addressed. By constructing a prototype that combines a monoscopic multi-focal plane HMD and stereo eye tracker, interaction with virtual elements such as text or buttons can be facilitated by measuring eye convergence on objects at different depths. This can prevent virtual information from being unnecessarily overlaid onto real world objects that are at a different depth, but in the same line of sight. The prototype I built was then used in a series of experiments testing the feasibility and limitations of such interaction. Despite only being presented with monocular depth cues, participants still had the ability to correctly select virtual icons in near, mid, and far virtual planes in 98.6% of cases with this algorithm. Additional experiments were carried out with two commercial single-plane monocular displays, showing that the algorithm can quickly distinguish between a virtual reading task and real world environmental gaze, and can be used to automate text control functions.

While depth based eye tracking provided a good solution for reducing distracting content in optical see-through displays, similar problems still remained for video see-through augmented reality. For example, most video see-through displays with a wide field of view are either bulky, lack stereoscopy, have a limited field of view, or must be manipulated manually. Functionality to improve a user’s field of view or vision such as peripheral vision expansion or telescopic ability can be incredibly useful, but lacks a natural and unobtrusive method for engagement. To address these problems, I propose a vision augmentation display that uses a unique, modular hardware framework that allows for on demand reconfiguration of vision augmentations and engagement of augmentative functionality through integrated eye tracking. Three major iterations of vision augmentation prototypes are implemented to provide one to one see-through AR capability, peripheral vision expansion, and binocular telescopic functionality. The final prototype includes integrated eye tracking to allow users to engage vision modifications in real time and in a hands-free manner. I then tested a number of different ways of displaying the vision augmentation streams through comprehensive experimentation. I first evaluated the initial vision expansion prototype for its ability to incorporate peripheral information into the virtual field of view. Next, I tested three different eye engagements and five visualizations for augmentative data, and then carried out a comprehensive analysis of engagement classification accuracy, eye movement, head movement, visual acuity, time to completion, improvements over time, and calibration error generated by reconfigurations of the modular system. Solutions to the challenges I discovered

during the prototyping process are discussed in depth, such as compensating for misalignment between the expanded periphery and undistorted central field and dealing with unnatural lateral translations in telescopic video streams.

In summary, this thesis provides three primary contributions, including a dynamic text management system to improve visibility of both content and environment, a multi-focal plane interface that can reduce interaction requirements for removing distracting content, and a modular framework that improves both flexibility and control of vision augmentation devices. In addition to contributions in both hardware and software, experiments are also conducted and analyses carried out to verify the effectiveness of each system.

Keywords: Head mounted displays; safety; view management; eye tracking; augmented reality, interaction; augmented vision.

Contents

CHAPTER 1

Introduction

Since the start of augmented reality (AR) research, managing content in the user's field of view has been a challenging issue. Content must meet a number of requirements, including viewability, readability, minimal invasiveness, proper occlusion, and low user distraction. Whatever the type of content, safety and usability are paramount when using head mounted displays (HMDs) such as that shown in Figure 1. For example, if a user displays several lines of text through an HMD, he or she will already begin to experience a number of unwanted effects. The text may be overlaid onto objects of concern in the environment, oncoming traffic may be occluded resulting in decreased reaction time, and the focal point of the text may be different than if the user were looking straight ahead while walking, as can be seen in the lower right image in Figure 1. While researchers have made many attempts to solve these problems through different strategies, many challenges still remain, and new ways of displaying content, such as vision augmentations, create new problems that must be overcome.

1.1 Properly Managing Content

It is first important to outline what kinds of information exist within the virtual world, what kinds of devices are used to display which information, and how these types of information

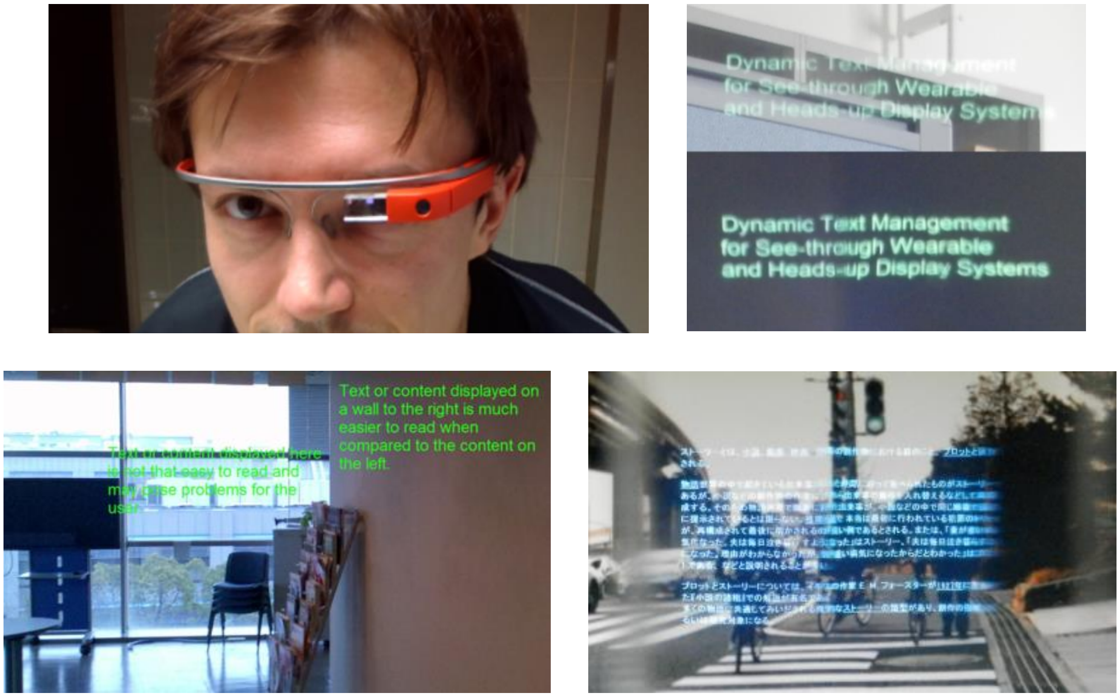

Dynamic Text Management for See-through Wearable and Heads-up Display Systems
Dynamic Text Management for See-through Wearable and Heads-up Display Systems
Text or content displayed here is not that easy to read and may cause problems for the user
Text or content displayed on a wall to the right is much easier to read when compared to the content on the left.

Figure 1 Images showing an HMD (top left) simulated text readability problem (bottom left), limited visibility of text over a light, textured background in comparison with a dark, uniform one (top right), and text overlaid onto a street crossing in an unsafe manner (bottom right).

need to be managed. As such, the following section covers typical problems with overlaid content, information classification, content orientations, and several other pertinent examples of management. The first and likely broadest type of information to classify is that of virtual information in general, which falls into augmented, mixed, and virtual realities, thus forming three types of reality spaces. Each of these well-known categories has a number of both shared and independent subtypes of information, each with its own set of problems (Rolland et al., 2000).

1.2 Overview of Problems in Augmented Reality View Management

Although there are numerous problems within the field of AR, several are specific to view management. While problems like delay and screen resolution are innate to the type of hardware or display technology used, problems like clutter, invasiveness, and limited FOV can potentially be solved by view management and viewspace manipulation strategies. These types of problems are outlined in more detail below.

1.2.1 Visual Clutter

The abundance of virtual information within a confined viewing space is referred to as **visual clutter**, much like a child's room may look when full of toys. Clutter results in a number of problems for the user such as decreased visibility of both real world and virtual information and increased interaction requirements to bring desired information into view, as shown in the upper left of Figure 2.

In the case of mobile augmented reality, clutter can occur in environments where information is dense, for example when presenting a number of building names and descriptions as labels to a user in an urban environment. When e-mails, notifications, and messages are also added to the user’s virtual viewing space, clutter can be further increased and restrict the user’s view of the real world. Too much content can also result in confusion or distraction, and may result in unwanted esoteric motion perception if presented in the user’s peripheral vision. Merging two fields of view can also be considered a type of clutter, since one FOV may occlude an important part of the other.

Invasiveness is also a challenge, and can be considered as both a type of clutter and as a general visibility problem. For example, when content is present and visible, it may be a distraction to the user or may interfere with everyday activities. Much like wearing swimming goggles all day would interfere with some activities, improperly placed content can be invasive and become annoying if it is not well managed.

1.2.2 Visibility and Readability

Much like clutter, there are a number of factors that can affect the **visibility and readability** of virtual text. Although display types will be described later, it should be noted that visibility and readability problems due to lighting and view space vary greatly between devices, especially optical vs. video see-through displays (Rolland et al., 2000). In general, incoming light and background texture can greatly affect the readability of text, and can interfere with realism of a particular augmentation, as can be seen in the bottom left and top right images in Figure 1. Conversely, even a single block of text can interfere with visibility of the real world. In some cases, these problems can be solved by increasing brightness or modulating background image, but different displays require different solutions.

1.2.3 Safety Concerns: Distraction, Limited Attention, and Occlusion

One last issue, and perhaps the most important for augmented or mixed reality, is that of user **safety**. This is particularly true for mobile or outdoor AR, where traffic, obstacles, or hazardous situations may potentially cause harm to the user if his or her visual field is restricted. A good example is shown in the upper right of Figure 2, which shows a social networking application occluding the user's view of traffic. Both occlusion of vehicles and differing brightness can significantly affect the user's perception of the real world. Although many false stereotypes, such as complete distraction from the real world, exist about the dangers associated with augmented or mixed reality views, many concerns are real, and can potentially be solved by view management strategies.

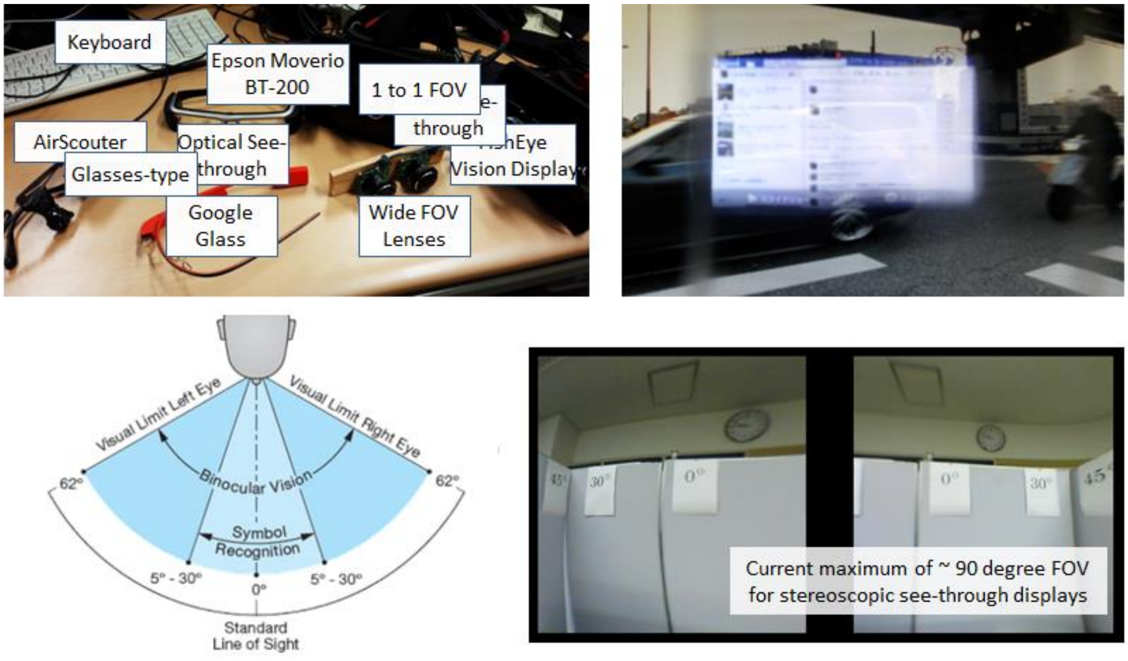


Figure 2 Problems that affect safety of mixed and augmented reality applications such as clutter (top left), occlusion (top right), and the limited field of view of a video see-through converted Oculus Rift (bottom right), with a diagram of the human field of view for reference (bottom left, Extron, 2015).

1.3 Influence of Information Type

There is a strong relationship between the type of virtual information displayed and the resulting problems users face in many AR applications. For example, a small, semi-transparent icon may not interfere with a user’s day to day activities, but may be difficult to see. On the other hand, a large, bright notification or message may be easy to see, but may occlude important information in the environment and endanger the individual’s real world activities. One distinguishing feature of content that I define in this thesis is that of user or environment centricity.

1.3.1 Display Medium

One of the most significant factors that affects view management problems in AR is the display medium used to present information. Most AR setups include either a head mounted display (HMD), tablet PC, mobile phone, static monitor with camera, or some combination of these devices. Due to the mobility and rising prevalence of head mounted technologies such as Google Glass, the Epson Moverio, Vuzix video see-through displays, and systems like Ovrvision for the Oculus Rift, this thesis primarily focuses on view management techniques for HMD systems. However, many of the strategies presented here can be adapted to tablet or phone systems employing the “window on the world” AR perspective, or telecommunication applications such as Skype. Head mounted displays primarily include two types, optical see-through and video see-through. Each display comes with a very different set of problems.

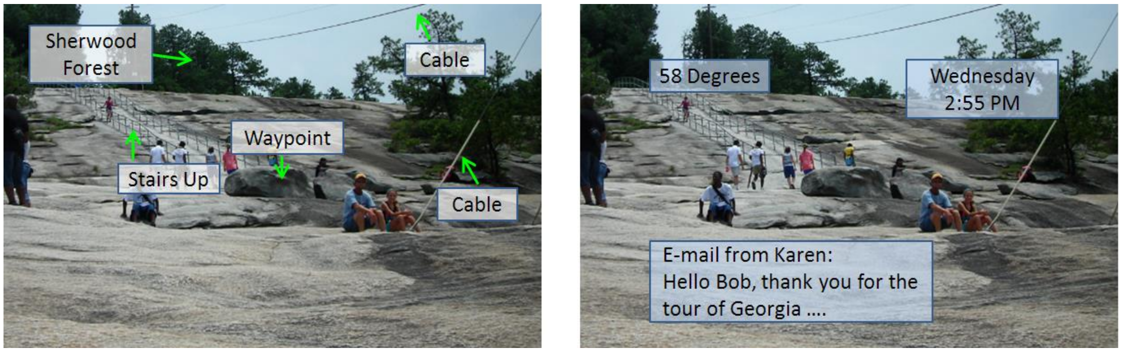
Optical see-through displays such as the Epson Moverio and other head-up displays (HUD) mainly suffer from problems with visibility and limited screen space. For example, in order to effectively use this type of display in a constantly changing, dynamic environment, variables like lighting conditions, human or vehicular obstructions in a user's path, and scene variation must be dealt with effectively. Because information is often presented directly in a user’s field of view, his or her attention to the environment can be reduced when concentrating on content. Moreover, personal information is often affixed directly to the display instead of the background environment, which can be unnatural and result in fatigue. Additionally, inconsistencies between content on the display and the real world can potentially confuse or distract the user, thereby decreasing safety and awareness when mobile. On the other hand, video see-through displays suffer from different safety related problems, such as limited field of view (FOV). Whereas users usually have a view of both the real world and virtual content at the same time in a typical optical see-through display, immersive video see-through displays such as the Augmented Reality Oculus Rift (Steptoe et al., 2014) often occlude large portions of the user’s peripheral vision, which can partially reduce or completely eliminate peripheral vision. Reduced peripheral vision poses an immediate danger to the user, since cars or other obstacles that would normally be detected with esoteric motion perception are no longer visible. This is a severe problem in mobile augmented reality since users of the devices may employ AR systems in heavy urban environments. In addition, delay, meaning the time it takes from the first light photon hitting the camera sensor to the last photon hitting the user’s eye, amplifies the previously mentioned problems since reaction times to events in the real world are further reduced. As mentioned above, the fact that information is presented in the user's direct field of view and the fact that the field of view is reduced are two main problems with current mobile HMD systems displaying AR content. As such, a large portion of this thesis is dedicated to addressing problems in an unobtrusive and non-invasive manner.

1.3.2 Environment Centric Versus User Centric Information

One type of information, which has likely received most attention to date by researchers, is **environment centric information**. This refers to the direct augmentation of an object in the environment, such as adding a label to a building or overlaying a digital character onto an AR marker. In prior AR research, environment centric or environment orientated information is likely the most commonly studied. In this case, information is oriented as if it were located in the real world. Text or content in this case is often attached to or overlaid onto an environmental object. Other examples include 3D registered building labels, waypoints, or geospatially located information. Labels of virtual items located in a game world such as uncollected weapons or navigation points are also environmental, in contrast with life, reticle, or currently wielded weapons, which have traditionally been display centric.

On the other hand, **user centric information** has a looser relationship with the environment, and has received little attention to date, despite being one of the most common types of information displayed in current applications for displays like Google Glass or the Epson Moverio. Rather than being closely related to an environmental object or location, this type of information is central to the user, and is often location independent. Good examples include e-mail, private notifications, reminders, video, and other types of messages. Unlike environment centric information, user centric information has traditionally resided on the display screen, without having any attachment or registration to real world objects. Two examples of each type of information are shown in Figure 3. Whereas content in the image on the left is tied to a specific location in the environment, content on the right will typically travel with the user.

Although leaving information on the display screen has many advantages for interaction, placing user centric information in the environment makes sense for the purposes of convenience and viewability. For example, it would be incredibly useful to be able to place a virtual calendar in a convenient location in one's office. Instead of having to remove and interact with a phone or tablet every time, virtual information would simply be viewable at all times. In this case, user centric information takes on an environmental or location based component, blurring the lines between user and environment centricity. User centric information has received much less attention from AR researchers to date, so numerous problems such as poor visibility and invasiveness still exist, which are especially prevalent in mobile AR. Next, I will more clearly define some of these remaining problems in AR, and specify which problems are specific to which technologies and types of information display.



E-mail from Karen:
Hello Bob, thank you for the tour of Georgia ....

Figure 3 Images of simulated *environment centric* labels with leader lines (left) and *user centric* content (right) overlaid onto an image taken from a hiker's viewpoint.

1.3.3 Orientation

Another important feature of virtual content is its orientation. Different orientations present different advantages and disadvantages for a particular display and type of content. Moreover, content must be managed differently depending on orientation, like when choosing whether to use a 2D or 3D management strategy. Currently, there are four main types of orientation: environment, body, head, and display, as described below. Environmental oriented is almost always environment centric, whereas body, head, and display oriented are usually user centric.

Environment oriented has less relevance to display location in terms of view management, although the display viewing window functions like a "window on the world" (Milgram et al., 1995) for presenting correctly registered and geometrically projected content on the display. Note that labels presented to the user in 2D can also be environmentally registered, but do not always require projective geometry for orientation, as shown on the left image of Figure 3. A subset of environment oriented is sometimes called object oriented, or oriented around or about a particular object or set of objects, as shown in the left image of Figure 4 (Tatzgern et al., 2014).

A second type is **body oriented**, which usually refers to positioning relative to the user's chest, torso, waist, or some general metric for the entire body. The benefit of this body orientation is that the head can rotate within this space, which is more appropriate for personal workspaces, windowed content, and user centric information. The simplest version of this orientation is a spherical display scheme, where all windows are positioned orthogonally to the torso, at a distance equivalent to the focal plane of the display in use. This results in a spherical workspace that can be navigated via head movement. Although tracking a user's body and head can be challenging, another benefit of this orientation is that user centric information will travel with the user.

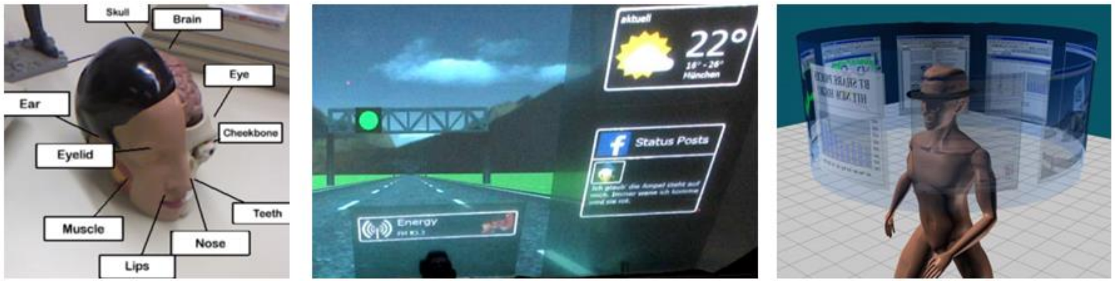


Figure 4 Images of different orientations showing object- (left, Tatzgern et al., 2014), display- (center, Häuslschmid et al., 2015), and head- (right, Billinghurst et al., 1998) oriented content.

Head oriented content is affixed to a location relative to a user's head, which is often the case with current commercial head mounted displays. A sample of this type of orientation is shown on the right of Figure 4. Since information is almost always located in the user's field of view, noticeability and awareness of virtual objects or pointers are relatively good when compared to environmentally registered content that may be located behind a user. One benefit of this orientation is that eye position can be tracked with very high accuracy within the display. This type of orientation is often combined with wearable eye tracking systems, which are becoming increasingly commonplace, because the position of the information is static relative to the user's head, and eye gaze based interaction can be more accurate.

Display oriented content is likely the most common type of orientation for current commercial wearable systems. Any information that remains fixed to the display screen and is moved relative to the display rather than another anchor point or registered object is considered display oriented. Unmodified application icons, game windows, notifications, non-augmented items, and head up display content, such as that shown in the center of Figure 4, fall into this category. Many 2D content management methods utilize display oriented content, even though that content may respond to or be arranged according to an analysis of the background scene.

In the event that a user is wearing a head worn display in a position affixed to his or her head, head and display oriented information are essentially the same. For devices such as tablets or phones that are decoupled from the head, head and display orientation are very different. Another important subset of display orientation is likely vehicle orientation. While a vehicle can be considered part of a user's environment, it is often moving with the user instead of the environment, and facilitates different types of information such as speedometer readings, navigation information, and warnings that may be registered or affixed to parts of the vehicle rather than external environmental objects.

Because a large amount of user centric information is displayed on frequently moving mobile devices such as Google Glass or car windshield displays, many of the adaptive display methods proposed in this thesis are geared towards head and display oriented content.

1.3.4 Types of Augmentations

Both **labels and annotations** are usually added for the purpose of describing or explaining an object in the real world or a virtual object placed in the real world. A very simple example would be placement of a virtual building name or address over the building in the real world, similar to the image on the left of Figure 3. As additional information is added, such as navigation cues, names, and descriptions of numerous items, content must be managed appropriately in order to provide the user the right information while still providing a clear view of the environment.

Much like labelling and annotation, **object overlay** refers to the direct modification of an environmental object with the intent to change the appearance of the object itself, such as in the left image of Figure 5. An example would be to place a virtual image of a portrait on the wall of an office, making it appear as if an actual portrait were attached to the wall.

A more recent advance in view management is the ability to remove elements from the real world, which is often referred to as **diminished reality**. This is a great complement to view management, since it can free up unused space in the real world for other more useful virtual information. An example is shown in the two center images of Figure 5 that demonstrates how an item can be removed from a scene to reveal important background information (Zokai, 2003). Removal of unnecessary content is becoming increasingly important since the space available to display augmented information quickly becomes smaller as the amount of information increases. Though systems that employ diminished reality can be incredibly useful, they must still be careful not to diminish important real world information such as stop signs or potentially hazardous obstacles when used outdoors.

Another core type of AR is that of **vision augmentation**. Rather than overlaying new virtual content, augmenting someone's vision involves enhancing or optimizing the user's view of the world. Traditional examples of vision augmentation displays include glasses, binoculars, night vision goggles, and infrared cameras. This type of augmentation is becoming available in many head mounted AR displays, which can give users telescopic ability, field of view expansion, night vision, or infrared imaging.

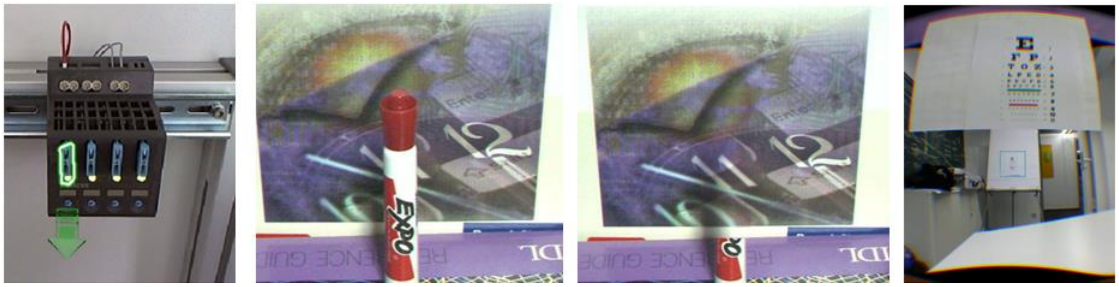


Figure 5 Augmentation types, including object overlay (left, Petersen et al., 2012), diminished reality (center images, Zokai et al., 2013), and telescopic vision augmentation (right).

Just like other labels or augmentations, vision enhancements must be managed so that users still have a reasonable view of the physical world and do not become disoriented.

1.4 Summary of Challenges Related to View Management

Of the previously mentioned concerns, I consider safety and usability to be the most important challenges to overcome. If a device is not safe to use, injury or even death may occur, and if the device is not easy to use or intuitive, its commercial success will be unlikely. With regards to safety, occlusion of environmental objects, invasiveness of content, and distraction are some of the most relevant problems to solve. With regards to usability, visibility, naturalness and intuitiveness are among the most related. This thesis will present a number of strategies that are adaptive to both the environment and a user's needs, which represent significant steps forward in the fields of mixed and augmented reality and vision augmentation. Next, I will conduct a comprehensive review of how the abovementioned problems have been solved up to now, describe what other researchers think, reveal what is still lacking with regards to safety and usability, and explain how my research builds on these prior works and contributes to the field.

CHAPTER 2

Overview of Prior Work and Strategies to Solve View Management Problems in Augmented Reality

Solving view management problems can be divided into a large number of sub-disciplines, but the most prior work can be divided into in three main categories. These include 1) traditional view management, which is designed to manage labels and annotations of both physical and virtual objects, 2) attentive interfaces, which provide functionality based on the state of the user or surroundings, and 3) view manipulation, which expands, improves or modifies the user's field of view in some way. These methods serve to reduce occlusion and clutter, improve visibility, and increase user awareness, all of which can improve safety and usability.

2.1 Traditional View Management

Traditional view management methods seek to manage text, labels or content by changing characteristics such as placement, color, and contrast. While a majority of these techniques are designed for labeling stationary objects, the techniques employed can also sometimes be used for managing display or user centric content.

2.1.1 Object and Environment Labeling Techniques and Studies

Because there is no single device that is utilized for any single application, a wide variety of view management methods and algorithms exist. However, regardless of device, object labeling is often conducted in a similar way. The labeling and placement of content around objects dates back to the early 90's, with a majority of work being done in the early 21st century. One main work that helped define the field of view management was by Bell et al., shown in the top left of Figure 6, which proposed the management of building labels and virtual elements around a marker based model (Bell et al., 2001). Content was moved constantly based on the rendering of environmental virtual objects and the addition of virtual labels to improve readability and minimize occlusion. They developed an algorithm to manage large numbers of building descriptions, images, and other annotative information that would otherwise overlap or clutter the viewing screen. By taking into account temporal continuity, occlusion, and screen constraints, the method arranged annotations in real time to maximize visibility. Like many other annotation methods, the algorithm is considered to be "greedy," is non-optimal, and does not background into account for text readability (Azuma et al., 2003).

Soon after, many studies appeared that studied readability and visibility in more depth, especially for virtual text. For example, Gabbard et al. conducted a study in 2005 using an optical see-through display to determine optimal readability for various surfaces such as brick, foliage, and pavement (Gabbard et al., 2005). Billboard configurations were also tested; for example, instead of using the environment as a background, a single color "billboard" was displayed through the HMD that functioned as a background, and text was displayed as if pasted onto the billboard. Participants sat still while gazing at a particular text style and background and times for reading text strings showed that green text and billboard drawing styles resulted in the fastest responses for readability. In order for text to be readable, the user should be able to clearly view text against various scenery and lighting conditions. In 2004, Leykin et al. used a pattern recognition approach to detect readable surfaces on static images using training data (Leykin et al., 2004). They used mean intensities of the text compared to the surrounding region to determine whether an area is readable or unreadable as well as a rating scale for readability. Their method was assessed as suitable for video see-through displays and they suggest that the technique is suitable for view management systems. A more in-depth study on virtual text in video and game environments was then conducted by Jankowski et al. in 2010 (Jankowski et al., 2010). This study presented results of positive and negative presentation styles in comparison with plain text and billboards for both reading time and accuracy, showing that billboard and negative anti-interference styles outperformed others.

In the early 2000’s, research was largely separated into methods that individually studied or focused on occlusion, readability, or visibility. Afterwards, a number of works appeared that were more comprehensive, and could solve a number of these problems all at once. Makita et al. calculate the optimal location for human labeling based on a person's location and camera analysis of the background image as shown on the very right of Figure 6 (Makita et al., 2009).

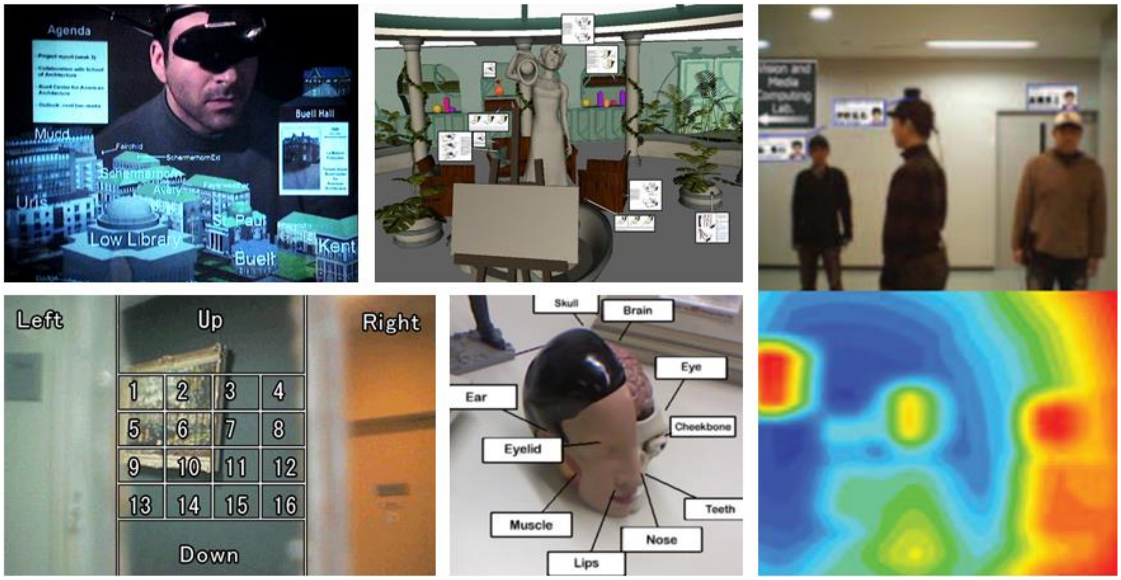


Figure 6 Images of label placement techniques including: view management of building labels in a marker based environment (top left, Bell et al., 2001), exploration of label management in an immersive video game environment (top middle, Stein, 2008), a tracker based system designed to label individuals in the environment (rightmost image, Makita et al., 2009), screen section based content management (bottom left, Tanaka et al., 2008), and 3D label management of an educational model (bottom middle, Tatzgern et al., 2014).

Methods proposed by Thanedar et al. and Tanaka et al. proposed using sections or divisions of the screen to manage text, shown in the bottom of left of Figure 6 (Thanedar et al., 2004, Tanaka et al., 2008). Other studies for text or label management exist, but focus on gaming, annotations of parts of 3D models, such as the Hedgehog labeling in the bottom middle of Figure 6 (Tatzgern et al., 2012), simulations, or other non-mobile or virtually submersive environments (Maas et al., 2006, Wither et al., 2009).

2.1.2 Outdoor View Management

Once AR devices were portable enough to be taken outside, a number of view management methods appeared that were designed to manage placement of outdoor content. One such method was developed by Grasset et al., which utilizes visual saliency and edge analysis to improve display of various annotations with historical or landmark information in urban environments (Grasset et al., 2012). This paper studies techniques designed for AR browsers, which can be deployed on a variety of HMD or handheld devices. Some of the other previously mentioned studies by Gabbard et al. and Leykin et al. were also influential on outdoor AR since they studied the readability of text in outdoor environments. Most recently, studies have been conducted on readability in industrial environments in which users may experience a variety of different lighting conditions (Gattullo et al., 2015).

While these algorithms function well for the designed purpose of improving visibility, the user studies are conducted on static images and are only designed for labels attached to objects in the environment. The methods above are sufficient for managing occlusion and visibility around a specific object or location, but do not adequately address user centric text such as e-mail or notifications. Because the nature of mobile, user-centric data is fundamentally different, different management strategies are required. More specific examples are discussed in Chapter 3.

2.2 Attentive Interfaces and Gaze Tracking

Another major field related to safety and usability in AR is that of attentive interfaces. A majority of these interfaces employ some type of user-state recognition, but the type of recognition that is most useful for HMD based AR is likely eye tracking. Eye tracking has a number of advantages over other sensor or activity based sensing mechanisms because of the proximity and direct integration of the tracker into the HMD itself. In many cases, the eye tracking camera or cameras are located inside the HMD, the occurrence of which has been increasing in recent years. Research on related attentive interfaces can be subdivided into two main areas: 1) the study of depth perception and eye movements, and 2) the study of gaze for use with attentive interfaces in AR applications. Both of these areas contribute to the management of virtual content and have the potential to improve user safety by increasing visibility and awareness.

Initial work in the field sought to influence human perceptions of virtual objects in a single focal plane in order to more accurately reproduce digital content. For example, Uratani et al. attempted to use depth cues in a video see-through display to alleviate the depth ambiguity problem in 2005 (Uratani et al., 2005). A similar study by Swan et al. studied depth judgments in an optical see-through display, emphasizing the importance of depth cues for depth judgments (Swan, 2006). Studies have also been conducted with a static 3D displays, such as that by Liu et al., which evaluated perception of accommodation, finding that focus is a viable depth cue for 3D displays in the monocular case (Liu, 2010). 3D gaze has also been proposed as a method for controlling prosthetic devices (Abbott et al., 2012). More recently, a number of studies on depth were conducted, the first of which tested a wide field-of-view HMD and measured user perceptions of Landolt C-rings in retro-reflective screens. Results showed that perceptual accuracy diminishes with distance, especially when only presented with a monoscopic image (Nguyen et al., 2012). Lang et al. studied the relationship between depth and visual saliency, showing that measuring depth can be used to improve saliency models (Lang, 2012). Though the work by Lang et al. was not a study on user perception, it further motivates the use of depth for improved interaction. Research also shows that eye movements play a significant role in human activities, and that fixations often occur prior to individual actions (Land, 2001). This also emphasizes the need for an attentive interface that can automatically provide a clearer view of a user’s environment.

While many eye tracking interfaces employ line of site for interaction, the results are limited to a 2D space, which is not often ideal for AR applications. Gaze has long been studied as a method for interaction, but only due to the recent developments in display and eye tracking technology have 3D displays, gaze depth, and vergence been considered for interaction (Chang, 2011, Lang, 2012, Woods, 2003). One of the first attempts at using gaze and depth for interaction in a static 3D display was conducted by Kwon et al. in 2006. They set up a system using a parallax barrier type stereo display positioned 84 centimeters away from the user, and were able to estimate depth from a user’s gaze toward virtual darts in one of 16 different regions of the screen (Kwon, 2006). Another application by Lee et al. used gaze and blink interaction for annotation tasks in a marker based AR workspace, though the display only utilized a single focal plane and focal depth was not considered (Lee, 2010). Pfeiffer et al. have also developed a 3D gaze tracking system that allows users to interact with elements in a virtual reality display (Pfeiffer, 2008). All of the studies mentioned above that utilize 3D gaze assume that binocular depth cues (an image presented to both eyes) will be present. However, very little research has been conducted on 3D gaze in monocular displays, and accuracy of gaze tracking when binocular cues are not present is still largely unexplored.

With the appearance of 3D gaze or depth based interfaces, researchers have also begun creating attentive interfaces so that the users of AR interfaces are less burdened with manual interaction. This requires a balance between the interface, intention, and human-interface communication in order to reduce interaction requirements. In addition to general structures and cognitive models for such interaction such as those proposed by Ferguson et al., more specific attempts at intelligently using gaze have been developed recently, such as the method for monitoring human attention by Ki et al. and the wearable reading assist system developed by Toyama et al. (Ferguson, 2011, Ki, 2007, Toyama, 2013). Prasov et al. highlight the importance of gaze based intelligent user interfaces for conversational agents, stressing that gaze plays an important role in floor management, grounding, and engagement (Prasov, 2008). Dalama et al. explore the use of an AR display equipped with an eye tracker for monitoring the cognitive state and attention of a visitor in art exhibitions (Dalama, 2012). Similarly, Qvarfordt et al. suggest that eye gaze can function as a mode of communication between human-computer interfaces (Qvarfordt, 2005). As described above, extensive design of attentive interfaces have been developed for binocular displays, so we wanted to build a more robust interface that can track 3D gaze in monocular displays, and test whether accommodation still functions as a good enough depth cue to facilitate interaction.

2.3 View Manipulation

As wearable displays have become lighter and AR technology has improved, the number of display systems developed to augment or improve human vision has also risen.

2.3.1 Changing Field of View

One initial attempt to design a display that utilized an optical augmentation to improve vision was by Harper in 1999 to assist with low vision rehabilitation (Harper et al., 1999). Soon after, Birkfellner et al. developed a variscope (operating binocular) to assist surgeons by incorporating a virtual plane into the surgeon’s real FOV at the same focal distance in 2002 (Birkfellner et al., 2002).

There have also been many attempts at improving the FOV of a head mounted display, such as the early work of Yamazaki et al. in 1999. They prototyped a prism based display that offered a 51 degree wide FOV (Yamazaki et al. 1999). Subsequently, a number of other design guidelines and display prototypes were created that used mirror and lens systems to expand the physical FOV to the periphery (Chen et al., 2002, Shum et al., 2003). In 2006, Nagahara et al. developed a display that converts the image from a 360 degree catadioptric camera system into two stereoscopically aligned images (Nagahara et al., 2006). These images, which compensate for distortion, are subsequently projected onto two hemispherical lenses, and provide a near 180 degree field of view. Another recent attempt to accomplish a wide FOV using projective displays was carried out by Kiyokawa. This display was developed using hyperbolic half-silvered mirrors in combination with a retro-reflective screen, which gives users optical see-through capability (Kiyokawa, 2007). Both designs by Nagahara et al. and Kiyokawa are relatively bulky, and require separate projectors and mirrors for each eye.

A similar display proposed by Ardouin et al. in 2012 also uses a catadioptric camera to compress 360 degrees of viewing field into a 45 degree FOV display (Ardouin et al., 2012).

Unfortunately, this introduces significant distortion into the user's binocular vision, and only a short quantitative experiment was carried out. To my knowledge, the most recent attempt at providing an expanded field of vision is that of Fan et al. in 2014 (Fan et al., 2014). They present a single 100 degree wide field of view camera image to both eyes (biocular view). Instead of a user being able to view his or her peripheral environment, a number of different indicators are blended into the displayed image to indicate objects of interest.

Finally, there have been several displays that are designed for telescopic viewing of the environment, which I also address. The first was by Lintu et al. in 2006, who developed an AR system that augmented views through an astronomical telescope to provide educational information and visual aids (Lintu et al., 2006). More recently, Oskiper et al. developed a set of AR binoculars in 2013, which allow a user to view augmentations through a device with a standard binocular form factor (Oskiper et al., 2013).

Most past studies on virtual peripheral vision in wearable displays have been limited due to physical restrictions of display technology. However, a number of studies are available that examine various projected objects or modified physical peripheral views in non-virtual environments. Human peripheral vision has been very widely studied, with one of the first relevant studies from Brandt et al., who showed that rotations of the periphery result in a perceived self-rotation (Brandt et al., 1973). This type of perceptual study has been extended into the virtual domain, such as the work by Draper et al., which showed that changes in scale can lead to simulation sickness in virtual displays (Draper et al., 2001).

More recently, researchers have begun to consider virtual displays for the modification of the periphery. For example, Vargas-martin et al. used an HMD to add peripheral information to the central field of view to help patients with severe tunnel vision (Vargas-martin and Peli, 2002). A more recent study by Loomis et al. in 2008 studied perceptions of gaze in human peripheral vision. It was discovered that, to some degree, humans can determine the gaze direction of an onlooker despite the fact that the onlooker's face is in the periphery (Loomis, 2008). Even more recently, the predator-prey vision metaphor has been proposed as a method for modifying the periphery by varying the camera angle to simultaneously increase the peripheral FOV while decreasing the binocular FOV (Sherstyuk et al., 2012). Annotation discovery rate has also been studied in wide FOV optical see-through displays by Kishishita et al. (Kishishita et al., 2013). This provides further evidence that effective use of both binocular and peripheral view spaces is essential when users need to notice objects beyond the binocular field of vision.

2.3.2 Engaging and Transitioning Augmentative Data

Much like augmentative displays, transitions and visualizations for augmented data have also been well studied. A number of works explore methods for addition of virtual augmentations. While the virtual additions are not vision augmentations, these works explore methods for merging, transitioning, and integrating augmented data, which have contributed to the general design of AR visualizations. Since the beginning of the 21st century, a seamless, consistent, and non-invasive view of augmented information has been the topic of much research. For example, in 2000, Kiyokawa et al. proposed a method for blinding the user to any scale manipulations during cooperative work (Kiyokawa et al., 2000). In a similar way, two of the visualizations in ModulAR, a display prototype proposed in this thesis, utilize transparency and reduction in scale for translations for a more seamless interface. Smooth transitions from one virtual scene to another are also important, as outlined by Billinghurst et al. in the Magicbook (Billinghurst et al., 2001). The system emphasized the importance of continuity when transitioning from one view to another, while taking Milgram's Reality-Virtuality continuum into account (Milgram et al., 1995).

In addition to transition-based methods, a number of works propose sub-windowed or lens-based methods for visualizing data. For example, Mendez et al. have proposed a context sensitive magic lens method for visualizing see-through or x-ray vision data (Mendez et al., 2006). These views can be used to more intuitively navigate models and visualize the internal structure of objects. Research has also been conducted for outdoor environments to improve spatial understanding of large scale scenes viewed from different cameras (Veas et al., 2010). Experimental findings showed that preference and performance can vary widely between visualizations, which served as further motivation for us to test different ways of merging various real-time streams of vision augmentation data. The idea of augmenting vision using a marker based "magnifying glass" system was actually proposed by Rekimoto in 1995 (Rekimoto et al., 1995). Instead of implementing an optical magnifier or zoom functionality, this work sought to incorporate pre-stored information into the user's FOV by recognizing markers in the environment, and subsequently presenting a larger virtual note on a hand-held monitor. More recently, virtual lens systems have been studied in more detail for the purposes of looking through to another "dimension" of data by Looser et al. (Looser et al., 2004). Even more recently, direct vision expansion using head tracking in a VR environment was conducted by Yano et al. for the purposes of modifying or magnifying the current FOV (Yano et al., 2014).

These studies provided additional motivation for us to explore vision augmentation for the purposes of AR. In addition, the telescopic and fisheye camera-lens systems are physically coupled to the user's head movement, so magic-lens based views, transitions, and transparency need to be managed differently.

2.4 Unsolved Problems

Because of the nature of user centric content, current view management methods and label placement algorithms are not enough to manage mobile content. Although some researchers have developed systems to manage mobile information, these works are either limited in scope, require prior knowledge of the 3D structure of the environment, or are designed for post-processing. Due to these limitations, a more adaptive and flexible system is necessary for managing content.

Secondly, while view management algorithms, including the algorithms proposed in this thesis, may be effective for managing the placement or visibility of text, there are times when a user may need a clear view of the environment. This is especially true for outdoor mobile augmented reality, where augmentations or virtual content can result in a direct obstruction of potentially hazardous objects. Current attentive interfaces are primarily developed for use with static displays, or are designed for selection and manipulation rather than removal of distracting items. Because of this, the need remains for an interface that can assist users in real time, especially for monocular displays that may reside in the central field of view.

In addition to the occlusion of environmental objects by virtual content, occlusion due to FOV limitations also poses a threat to the user's safety. Though many systems up to now have been designed to expand or manipulate FOV, many are bulky, expensive, or lack a binocular component. Additionally, most devices that expand or modify FOV are fixed form factor, cannot be changed on the fly, and are difficult to duplicate for other researchers. Moreover, augmentations are usually displayed in an "always-on" fashion, meaning users lack an effective method for engaging or disengaging the augmentations in real time.

This thesis seeks to address the problems mentioned above through the use of new adaptive content management and on-demand FOV manipulation techniques.

2.5 Contributions of This Thesis

Along with the overview of prior work presented above, the contributions of this thesis can be separated into several components, including hardware, software, and experimentation. Throughout the thesis, new display technologies and HMD prototypes are described, along with the software used to implement virtual components and manage content in the displays. Thorough evaluations are then conducted on all of the various view management techniques to study how humans think about and perceive virtual content and how these new prototypes perform for a number of different use cases. These contributions are summarized below.

2.5.1 Hardware

The first hardware contribution of this thesis is a multi-focal plane monoscopic display prototype designed to facilitate interaction through eye tracking. By classifying the focal plane in which the user is engaged, text or content can be quickly removed when the user changes focal depth. This strategy, as described in Chapter 4, can prevent virtual information from being overlaid onto real world objects that may present a hazard to the user by taking advantage of the user's focal point. The display itself is composed of three horizontally aligned AiRScouter HMDs, a 3D printed connector, and an SMI eye tracking apparatus. Unlike other eye tracking integrated systems that rely on binocular cues, eye tracking in this prototype is conducted using the convergence that results only from the accommodative cues in the monoscopic display.

I also introduce the Fisheye Vision and ModulAR video see-through display prototypes, which allow for vision expansion and eye-controlled vision augmentations, and are described in detail in Chapter 5. The final prototype consists of an Oculus Rift DK2, SMI Eye Tracker, and various camera-lens systems. Unlike previous designs, the system allows for configurable hardware augmentations and minimally disruptive software visualizations that can be freely engaged via intuitive eye movements. Modularity is achieved by using LEGO building blocks to interchange hardware modules.

Lastly, methodology for calibrating and integrating view management methods for both the Fisheye Vision and ModulAR displays is described in detail. Additionally, since I utilize displays such as the Brother AiRScouter, Google Glass, Epson Moverio, and Oculus Rift, the logic behind integration of each software contribution is described in conjunction with the hardware. Moreover, integration of the software methods described below are described in detail in conjunction with the hardware.

2.5.2 Software

The first algorithms presented in this thesis are designed to manage the placement of virtual content in the user's immediate field of view. They are unique in the sense that they are designed to deal with user centric content such as e-mail, messages, and notifications that are not affixed to predefined points in the environment. Two primary placement algorithms are presented, including Dynamic Text Management, which is designed to maximize text visibility by moving content along the user's travel path, and Halo Content, which is designed to prevent text from becoming invasive in personal conversations or interactions, as described in Chapter 3.

Additionally, methodology for conducting focus (accommodation) based eye tracking is introduced in Chapter 4. These algorithms take advantage of the natural convergence of the eye onto objects presented in various monocular displays, and are designed to compensate for the fact that the accuracy of tracking in a monoscopic display decreases significantly in comparison with displays that provide the user with binocular depth cues, primarily vergence.

I also introduce new modified computer vision algorithms and presentation methods that are designed to expand or improve a user's field of view in video see-through displays. One of these computer vision algorithms, used for the Fisheye Vision Display, introduces a new way to combine standard see-through AR with a compressed peripheral component without disrupting the continuity between central and peripheral vision. Building on this concept, I then introduce a number of presentation methods for introducing streams of augmented vision within the ModulAR framework, such as a telescopic (zoomed) view, into a standard see-through view, which are presented in Chapter 5. Various presentation styles such as sub-windowed methods and snapshot views are introduced and studied in depth.

2.5.3 Evaluation

A number of comprehensive studies are also presented that verify the accuracy, usability, and benefits of both the hardware and software contributions detailed above. These evaluations include initial awareness and performance tests to compare and contrast HMD to smartphone use. These tests are followed by the development of the Dynamic Text Management algorithms in Chapter 3, which are tested to compare the placement choices of the system with human choices in both static images and videos, revealing a 70% similarity in the system's and humans' placement tendencies. This experiment is followed by the development of the Halo Content algorithm, which is evaluated for its ability to reduce the invasiveness of content while still allowing for easy viewing of content. Results show that a 54.8% reduction in invasiveness is achieved in comparison to screen based layouts.

Next, the eye-tracking based attentive interface from Chapter 4 was evaluated on several different hardware devices. It was first tested on the multi-focal plane prototype, which showed that tracking of virtual content in a monoscopic display, though inaccurate, is still possible with approximately 98% accuracy if a focal depth classification is employed based on knowledge of the display's focal plane distances and content positions. This algorithm was then tested on two commercial devices, and though accuracy decreased with increased focal plane distance, we can still achieve greater than 90% accuracy for focal planes under or around one meter.

Finally in Chapter 5, the Fisheye Vision and ModulAR display prototypes are evaluated. The Fisheye Vision display was first evaluated to determine if it truly provides a 180 degree field of view, and results show that users can still view objects up to 180 degrees, but with a 27.5% overall reduction in the number of objects noticed, the accuracy of which varies with object size. The ModulAR system was then built and evaluated on a number of different aspects, including intuitiveness, fatigue, head movement, search and recognition accuracy, eye movement, and calibration error for switching camera-lens modules. Results showed that binary techniques resulted in the highest accuracy, but snapshot based methods resulted in less head movement. Calibration results showed that despite switching camera-lens modules a number of times, calibration error is limited to fewer than 0.46 degrees in almost all cases, requiring little to no re-calibration when interchanging or reaffixing camera-lens systems.

Smaller sets of focused experiments for various related applications are also explored, such as emergency navigation, mobile text input using an input prototype called the Torso Keyboard, and calibration of an AR memory augmentation system.

2.6 Integration and Modularity

This thesis also discusses the potential for integration of different elements into a single framework. Though the hardware and software contributions mentioned above are designed for different applications and different sets of hardware, many of the ideas and strategies presented are candidates for integration into a single framework. For example, optical see-through and video see-through displays are usually divided into different fields and said to be limited for certain applications. However, it is very likely that these displays will utilize attributes from each other. Both types of display already employ technologies like simultaneous localization and mapping (SLAM) to tracking purposes. (Reitmayr et al., 2007). In the future, it is very possible that we will see a merging of the two technologies into a single hybrid platform that can provide both optical and video see-through functionality on a per pixel basis using LCD and opaque display technology.

As such, it is important to discuss ways in which the methods in this thesis can be applied to other platforms and hardware. In addition to discussion methods for integration in each section, the discussion will also review what strategies can be applied to what hardware / platforms. Additionally, the idea of a modular device, especially one that has the flexibility to change its optical properties on demand, will be discussed in more depth. Much like standardization of computer parts has resulted in more flexible, easily upgradeable systems, this same kind of modularity will be important in defining future display technologies.

CHAPTER 3

Active Text and Content Management

As mentioned previously, the text and content management section of this thesis will primarily describe the Dynamic Text Management and Halo Content algorithms, as well as the pilot studies and thought processes leading up to their creation.

3.1 Introduction

In recent years, the world of mobile computing has seen a drastic increase in the development of wearable display systems. Google's Project Glass, Epson's Moverio, and the Silicon Micro Display are just a few examples of display systems that allow users to view content overlaid onto their immediate environment while mobile. This type of technology has various benefits, such as allowing virtual information to be displayed onto real world objects, giving a simultaneous view of digital content and real world environment, and faster access times to information. Although these devices can be incredibly useful and augment human ability, various studies on technologically induced intersensory conflict, readability, and fatigue show that it is still difficult for users to acclimate to content viewed through see through displays (Huckauf et al., 2010, Mon-Williams et al., 1993, Nelson et al., 2000).

Secondly, since text is typically viewed in a fixed location on an HMD screen, people, vehicles, and noise in the background of a user's path can interfere with tasks, especially with reading text. Similarly, in vehicle heads up display (HUD) systems, displayed content such as speedometer readings, directions, and map information can interfere with visibility. For example, the images at the bottom of Figure 1 shows the most common method for displaying text on an HMD, affixing it to the center of the viewing screen. Interferences such as reflected sunlight, non-uniformity, and color variation all interfere with reading text. This type of hindrance to reading speed has been shown on numerous occasions (Gabbard et al., 2005, Leykin et al., 2004). In addition to interfering with tasks, digital content that occludes real world objects can pose a threat to a user's safety.

3.2 Prior Work

Because of the fear of a decrease in visibility in wearable displays, a number of works exist that study the effects of virtual information on effects such as environmental distraction and inattentional blindness. Two studies on awareness by Liu et al. and Birnholtz et al. are closely related to the initial pilot studies in that they test a user’s level of awareness while conducting a concentration intensive task. The first work by Liu et al. studies general attention of anesthesiologists to significant events in their immediate surgical environment when presented with other relevant information through an HMD (Liu et al., 2009). Results showed that the HMD actually increased users’ access times to external information, increased the length of visual contact with the patient, and had very little effect on noticing important events, such as a sudden drop in blood pressure. The second study by Birnholtz et al. focused on the same problems of attention and awareness, but in a workplace setting using a PC and projected peripheral information (Birnholtz et al., 2010). The system projected icons onto a wall in the peripheral vision of a number of PC users, and found that compared with a visual representation displayed directly on the monitor, projected notifications can expand a user's awareness of relevant events. Other related studies test more specific types of distraction, but from these two studies alone, it can be assumed that information in the peripheral vision, presented by either an HMD or projector, can improve attention and awareness in cases where a user is stationary (Birnholtz et al., 2010, Liu et al., 2009). These findings provided motivation for us to conduct similar tests in an outdoor mobile environment, especially since mobile and dynamic situations are still relatively unexplored.

3.3 Pilot Studies

Although several assumptions exist on the differences in awareness and distraction between HMDs and smartphones, few formal studies have been conducted that make an objective comparison in mobile augmented reality. I started by setting up such a study so that I could get a better idea of the real differences and challenges of the two types of device.

3.3.1 Viewability and Environmental Awareness

With the concerns about environmental awareness and distraction in mind, the first experiment I carried out was designed to test general contextual awareness of an HMD compared to a mobile phone. Pedestrians conduct a variety of reading tasks on mobile phones on a day to day basis, so I decided to test how often a user would notice an object in his or her surrounding environment when walking along a set path. I tested users' awareness levels on a Samsung Galaxy 3 smartphone (4.8 inch display, 1280x720 pixels) and an Epson Moverio BT-100 HMD (23 degree field of view, 960x540 pixels).

3.3.1.1 **Setup**

In the experiment, two sets of 10 college students read a newspaper article while walking down a sidewalk. Each student was informed that they would be testing a new interactive application for reading newspaper articles. They were instructed to walk for approximately 100 meters while reading a designated newspaper article. 10 participants used the touch based mobile phone to interact with text, and the 10 other participants used the Epson Moverio BT-100 HMD. They were instructed to read as much of the newspaper article as possible and stop at the end of the sidewalk, as shown in Figure 7, where an experimenter would be waiting for them. What the participants did not know is that another experimenter would be waving to them from a location approximately 2 meters away from the walking path, marked as D in Figure 7. The experimenter started waving when the participant entered the designated waving area, labeled B in Figure 7, and stopped waving the moment he or she left the area, regardless of the participant's walking speed.

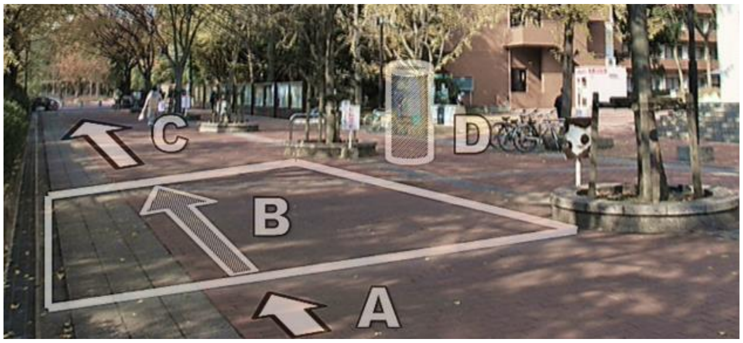


Figure 7 Experiment area showing A) walking path, B) walking path through waving zone, C) path toward finish, and D) location of waving experimenter.

Both sets of 10 participants completed the task to the end, and a survey was then given asking whether or not the participants had seen a waving person. A subjective survey using a 5 point Likert scale was also given, including the following questions: 1) Was the device easy to use while walking? 2) Was using the device scary? 3) Was it easy to view your surroundings?

3.3.1.2 **Results**

When asked if they had seen a waving person, results showed that none of the participants using the mobile phone had noticed the waving experimenter. In comparison, 7 of 10 participants using the HMD had noticed the waving experimenter. From this, I concluded that participants were to some extent more cognizant of their surroundings when using the wearable display compared to the mobile phone. A one way analysis of variance (ANOVA) on subjective results, where device was held as variable, showed no statistical difference between questions 1 and 2, (Q1: $F_{(1,19)}$=2.07, P=0.17, Q2: $F_{(1,19)}$=.026, P=0.87). However, there was an effect of device for question 3 (Q3: $F_{(1,19)}$=21.0, P<.01,) where the HMD group rated ease of viewing surroundings with an average of 3.6, but the phone group only had an average of 1.5.

This means that for reading tasks, a user's view of his or her environment is indeed better with an HMD and is also subjectively perceived as better than with a smartphone. I also received several comments that text on the HMD screen intersected with objects in the background, possibly decreasing reading performance. Based on these comments, I decided to conduct a second pilot to study user performance on an interactive task.

3.3.2 Follow up Experiment on Performance

3.3.2.1 **Setup**

From the first pilot, I hypothesized that A) additional environmental awareness may have decreased the ability of users to concentrate on reading and inferred that B) if possible, it would be much more effective to position text slightly away from the walking path so that text is still readable, but does not interfere with walking. A second experiment was conducted in a similar manner to the first, but designed to test A) from above through an attention intensive task. Instead of having two separate user groups, each of 7 participants conducted 4 tasks with both the phone and wearable display in a randomized order.

Participants, none of who had participated in the first pilot study, walked approximately 300 meters while playing a game involving dexterity and timing. The game involved repeatedly pressing a button when falling blocks on the screen intersected a stationary block, allowing us to measure time and accuracy. In addition, they also played the game while stationary to establish a baseline for performance. All trials were conducted on both the HMD and smartphone, resulting in 4 trials per participant, 2 inside and 2 outside.

3.3.2.2 **Results and Motivation for Dynamic Text Management**

Accuracy and time to completion were measured, and the phone were found to be slightly better for performance, but worse for task completion time. Results for smartphone vs. HMD were 94% vs 88% accuracy, 4.08 vs 5.43 pixel variance, and 224 seconds vs 221 seconds, respectively. Further informal experimentation suggested that device does not appear to have a large impact on performance. In both pilot studies, a number of participants stated that they could see more of the environment, but that other pedestrians and light interfered with tasks since they crossed the same viewing field where text was overlaid with the HMD.

It became obvious that this overlay problem was the most significant and that an automated solution was necessary. Many existing annotation systems attempt to manage text overlay by placing information onto fixed points in the environment, but do not have the ability to move text to new locations as a user travels.

3.4 Managing Text to Improve Viewability

In response to the concerns raised in the pilot experiments, I set out to improve mobile text placement in the environment from several different perspectives. Since users stated that they would prefer text located in more visible spots in the background rather than affixed to a single point on the HMD screen, I set a goal of moving text to desirable screen locations. As previous research has shown, text that is affixed to locations in the environment (rather than on the screen) can increase sense of presence and improve readability (Chen et al., 2004). I designed the system to address the issue of environmental text placement for e-mails, text messages, and user-centric text in general. Using camera tracking, the system automatically moves text to more visible locations on a user's walking path while mobile (Orlosky et al., 2013). Unlike prior methods, this system places text in dark, uniform areas in the environment in real time, and constantly maximizes visibility. The system setup can be seen in Figure 8.

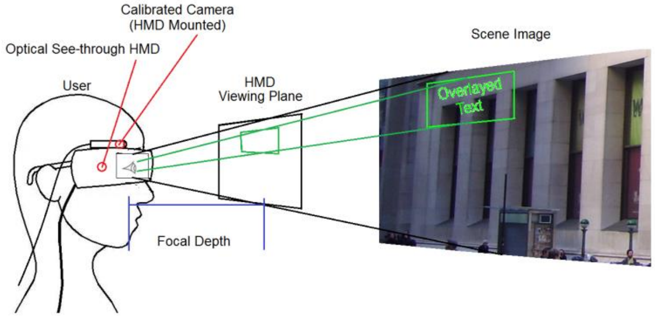


Figure 8 Image of the HMD setup, including mounted camera, focal plane, and a background image with overlaid text.

If a user could reach out, grab, and place overlaid text in a desired location, he or she might choose a dark or uniform surface where cars and pedestrians are unlikely to interfere. So, I sought to answer the question: How should content that is overlaid onto the real world be moved through one's environment? More specifically, how can text overlay and text movement be automated in real time? When choosing an appropriate region for text in the real world, a large number of factors play a part in that choice, affecting factors like readability, eye fatigue, and ease of use.

Video see-through displays can alleviate some of these problems by modulating real world images, but cannot solve all the problems completely, and are not ideal for all applications. On the other hand, optical see-through displays are often more suitable for vehicles or military applications for example, but have larger problems with readability due to luminance and inability to modulate or manipulate the background image. Because of these difficulties, the effects of background surface, lighting conditions, and other environmental interactions have been well studied (Gabbard et al., 2005, Leykin et al. 2004). However, the continued moving of content, such as e-mail and messaging, through one's environment have been largely overlooked, especially when users are mobile. In order to provide a new foundation for managing dynamic content and improve the usability of optical see-through HMD and HUD systems, I implemented an intelligent text management system that allows user centric text to move along a user's path in real-time.

3.4.1 Prior Work

Related research includes studies on readability of virtual text and visual preference and systems designed to improve the usability of wearable displays by moving overlaid content.

Though it is important for a HMD user be as aware as possible of his or her immediate environment, text on the HMD screen must be readable and placed in a desirable location in the environment. Prior to development of automated text and view management algorithms, a number of studies were conducted on text readability and visual preference using see-through displays (Jankowski et al., 2010, Scharff et al., 1999). Two studies in particular involve a wide range of experiments on readability of text overlaid onto different surfaces in static environments. The first of these studies by Leykin et al. builds a classifier to determine text readability based on texture properties and visual features. Textures included surfaces like cloth, wood, hair, cement, and mixed surfaces such as building facades (Leykin et al., 2004). Images of each texture with overlaid text were presented on a computer monitor and participants rated each case for readability. Another study by Gabbard et al. was conducted soon after that studied text readability using a see-through HMD. Experiment participants sat still and viewed text in a similar manner to Leykin's experiment, but looked through an optical see-through display in outdoor lighting conditions (Gabbard et al., 2006). The results showed that for static text, bright green provides for the fastest reading times and that text overlaid onto pavement, sidewalk, and foliage had the slowest reading times.

Several of the studies mentioned above suggest that the frameworks proposed could be used as part of an active management system (Gabbard et al., 2005, Leykin et al., 2004). Some active management systems exist, but only make attempts at managing annotations and text affixed to certain objects in the world (Maass et al., 2006, Thanedar et al., 2004). The question of how to appropriately manage and move user-centric text still remains unanswered. Furthermore, algorithms for detecting readable areas exist, but are not always implemented in real time, do not find the best area for readability, and provide no methodology for moving text from one readable area to the next (Bell et al., 2001, Makita et al., 2009, Thanedar et al. 2004).

As mentioned previously, in order for text to be readable, a user must be able to clearly view text against various scenery and lighting conditions. The study by Gabbard et al. showed that reading green text strings and billboard style displays resulted in the fastest responses for readability. Therefore, I chose bright green text for use in the method and attempted to fit text to areas that effectively function as billboards, which will be described in more detail later.

The most recent and closest work to dynamic text management is likely the annotation placement method developed by Makita et al. in 2009 (Makita et al., 2009). Via camera, this method estimates where annotations should appear by detecting humans located in real-world indoor environments, and subsequently places an annotation such as a name or image over an individual's body. The algorithm is penalty based, giving stronger likelihood to higher body locations such as the head or torso. It is implemented in real time, but does not take into account scene background or complexity for readability.

Other information management frameworks include more specific methods and classifiers for managing annotations in environments where space and screen limitations are present. Most of these methods are designed for gaming, advertising, labeling of individual parts of 3D objects, and other label placement (Maas, 2006, Makita, 2009, Thanedar, 2004, Wither 2009). Many other studies focus on text readability, but only do so in immersive virtual environments such as games, video, and 3D simulations. Although they provide useful insights about how to manage text in a simulated environment, they are not as relevant to real-world and mixed-AR applications.

Before continuing, it is important to restate the distinction between **environment centric** and **user centric** information presented to the user when using a see-through display. Similar classifiers exist that provide guidelines for annotation permanence (Wither, 2009) and whether head, body, or world is a better location for workspace (Billinghurst, 1998). This new classification is useful when determining whether content should be permanently fixed to an object in the environment or whether it should travel with the user. Though some types of information fall in both categories, the distinction is often easy to make.

The primary feature of user centric information is that it has a strong relationship between content and environment location. Examples include building descriptions, landmark information, billboard advertisements, and a majority of object annotations. The nature of this type of information is local to the object in the environment. For example, when displaying the name of a building, the information is most useful if affixed to the building itself. There is little need to move the information from its static, world-relative location to a new point in the user's field of view. A majority of current text placement research focuses on annotative and environment centric content (Bell, 2001, Maass, 2006, Tanaka 2008, Thanedar, 2004).

3.4.2 Framework and Algorithm Fundamentals

Keeping the previously mentioned challenges in mind, I designed this system to address them in several different phases. First, the system recognizes viable (dark, uniform) text locations in the real world using camera analysis. This analysis takes methods like Makita et al.'s a step further since it can be used in any environment without prior knowledge of the scene (Makita, 2009). It then affixes text to the most viable area, rotates the text to a stable orientation, and repeats this process from frame to frame. When a viable area becomes non-viable or leaves the user's field of view, the system moves text to a new area, maximizing the window of time in which the text is most viewable, and at the same time avoiding environmental interference. In each of the following subsections, I will describe the theory behind the framework as well as the details of how each component was implemented.

3.4.3 Selecting Viable Regions for Text

The first step in the framework is to find an appropriate location for text. Though many factors were taken into account, I primarily focused on detecting two scene features, darkness and uniformity, to find the best location. As Gabbard et al. and Leykin et al. both showed, lighter text on a dark, uniform background is one of the more visible display styles (Gabbard et al., 2005, Leykin et al., 2004). Out of light text colors, bright green (#00FF00) was chosen due to its contrast with colors of a majority of man-made structures, predominantly gray. Though placing dark text on a white background is also a highly readable display style, it is not currently feasible with optical see-through displays because of their inability to reproduce black due to transparency, though some prototypes can provide an occlusion mask to block out light (Kiyokawa, 2003). White "billboards" also result in frequent occlusion of other scene objects and decreased visibility, so I opted for placing bright green text over dark backgrounds as shown in Figure 8. Though bright green works in many environments, complement colors should also be considered, especially in environments with dense foliage or greenery or industrial settings (Gattullo, 2015). Because active modulation of text color is a relatively complex problem for dynamic environments, implementing a complement color based algorithm is not within the scope of this method. Lastly, based on a study conducted by Scharff et al. in 1999, text size was not determined to have a significant effect on readability, so an appropriate text size should be selected based on viewing distance and the dimensions of the HMD or HUD screen (Scharff, 1999). In essence, this method finds dark, uniform rectangular regions that resemble black billboards and then applies a best fit algorithm based on the dimensions of displayed text or content, which are both parametrized. To display e-mails or short messaging content, 8-10 pixel wide characters are used, resulting in between 40 and 55 characters per line.

Required angular height and width of the dark areas in the camera image are calculated based on the focal distance of the HMD viewing plane shown in Figure 8. Though selecting only dark regions or only uniform regions to display text is a valid method for finding readable areas, evenly weighting darkness and uniformity results in a larger number of unique regions in the scene to display text. A viability rating for darkness and uniformity is calculated at every point in the image, and a best *x*,*y* position is selected after rating all points in the image. I first calculate average pixel intensity for the whole image. Then, to rate a single point in the image, I use all pixels in the rectangle surrounding that pixel, sum grayscale intensities to calculate darkness, and take the standard deviation to determine uniformity. Rectangle size is determined by the dimensions of desired text. See Figure 9 for before and after images and the resulting heat map showing good points (darker = less viable, redder = more viable) and the best point in the image (bright green square). Text is then redrawn on the HMD viewing plane in a position calculated from and centered on the previously selected best point.

Algorithm:

```
WHILE camera is on
   CALCULATE average pixel intensity for current frame
   FOR each pixel in the current frame
      FOR all pixels in desired text display area (rectangular)
            CALCULATE relative darkness-to-intensity metric and uniformity (stdev)
            IF darkness + uniformity > previous viability rating
        THEN current pixel is new best point ENDIF
      ENDFOR
   ENDFOR
   REDRAW text on new best point
ENDWHILE
```

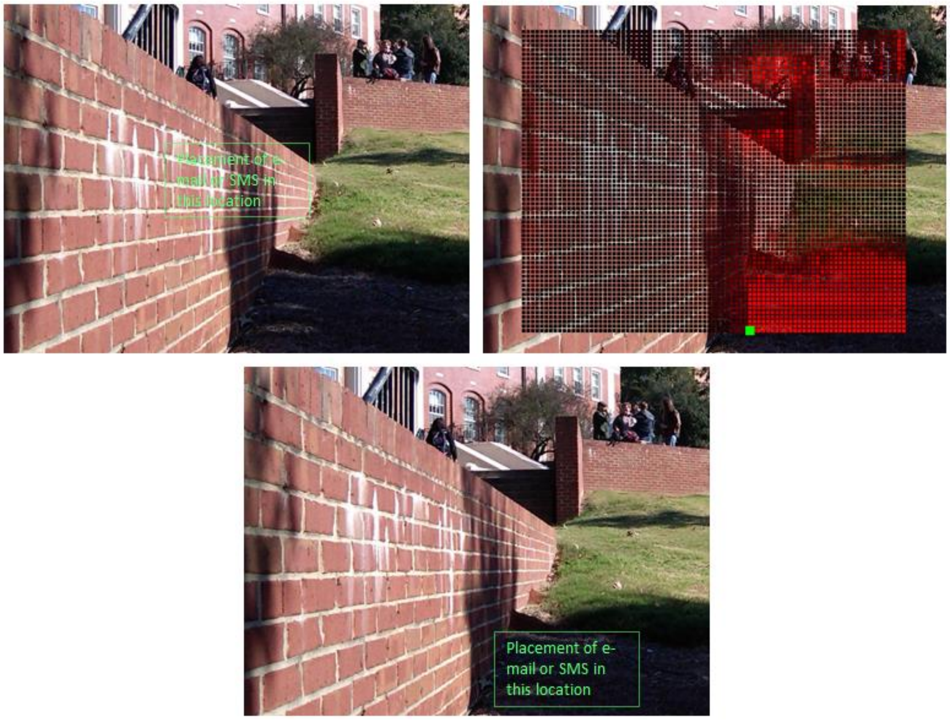

Placement of e-mail or SMS in this location
Placement of e-mail or SMS in this location

Figure 9 Images showing text placed in screen center, heat map showing viability analysis where darker is less viable, redder is more viable, and the best point is colored bright green.

Because humans and automobiles are not typically dark, uniform surfaces, this method also functions similarly to an object avoidance algorithm in addition to maximizing readability.

3.4.4 Content Movement

Since ideal regions for text often change significantly from frame to frame due to changing lighting conditions, text would sporadically alternate locations in the user's field of view if moved to a better ideal region every time. This type of movement is difficult for the eye to follow, so to maximize readability, I developed and incorporated a decision making algorithm that determines when and how to migrate text to a new location.

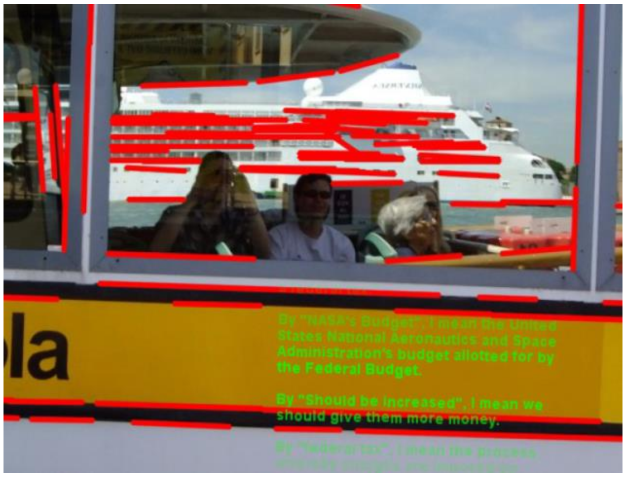
I use two primary schemes: 1) Move text directly to the next ideal location if the change in ideal location is only due to camera movement, and 2) only move to new, distant locations in the environment if the current location has become much worse than the new one. The results of these schemata are that 1) text remains in the same region in the environment despite a user's normal head movement, and 2) text is migrated to new viable regions in the environment only when the current region has become non-viable when compared to the new region. The difference between head movement and changing of viable location is computed via a predetermined threshold value. One more attribute is a running average of the last *n* positions, resulting in a smoothing effect when migrating text to a new region.

Algorithm:

```
CALCULATE expected maximum head movement
CALCULATE expected minimum distance for a change in viable locations in the environment
(non-head movement)
SET thresh to a pixel value between the above calculations
    WHILE selecting a new point
        IF distance to new best point < thresh
        THEN Current best point is new best point ENDIF
        IF distance to new best point > thresh AND new viability rating of best point /
        scaled distance to new best point > viability rating of old best point
        THEN Current best point is new best point ENDIF
    ENDWHILE
```

3.4.5 Text Orientation and Stability

Finally, in order to relieve technologically induced intersensory conflict, I built stabilization into the method. This means that text is rotated to an angle aligned with objects in the scene or to other features in the environment. This is most applicable to HMD screens since text is typically screen stabilized (fixed to the user's head position), which results in a disparity between real world and augmented information. Real world billboards or signs do not move synchronously with a user's head, so in order to counteract this disparity, I present two methods to align text with the environment and categorize situations in which each method is appropriate. Text is stabilized in 2D based on a study by Chen et al. that shows text overlaid in 2D HUD configurations is more readable, especially in dense or crowded environments (Chen, 2004).



By "NASA's Budget", I mean the United States National Aeronautics and Space Administration's budget allotted for by the Federal Budget.

By "Should be increased", I mean we should give them more money.

Figure 10 Lines detected by the Hough lines algorithm and resulting text that has been stabilized based on the average, weighted line orientation.

Gravity-stable means that text is always aligned perpendicular to earth's gravitational field and is also similar to world-stable, though position of virtual information changes gradually with user position (Billinghurst, 1998). Since this method can be accomplished with accelerometers, it is appropriate for noisy environments where scene analysis may be difficult or when a user's or vehicle's movement in the vertical plane is smooth. If a user is walking quickly or is in an environment where sporadic movements are necessary, noisy acceleration readings can result in poor stabilization.

A new method for stabilization and one of the contributions of this paper is region-stable alignment, which means that text is aligned to the surface or surrounding surfaces on which text is overlaid. From a user's point of view, man-made surfaces do not always appear to be aligned with gravity. For example, though an HMD screen is roughly aligned with gravity, buildings that are also aligned with gravity will appear to be at an angle againstvirtual information. If text is aligned over the buildings, it should be aligned parallel to the structure's apparent orientation rather than to gravity. Aligning to a single surface does not necessarily result in the best stabilization since angles of 3D objects can vary greatly within a small area. Chen et al. also showed that 3D aligned text is not necessarily the most readable (Chen, 2004). Stabilizing to the surface of an object also requires prior knowledge of 3D scene geometry. With my method, I can calculate a metric for region-stability that does not require this prior knowledge.

To find alignment of the region in the user's field of view, I start by extracting all detectable lines in the current frame using Hough's pattern detection algorithm (Hough, 1962), weighting their respective angles according to line length, and calculating an average angle for the entire viewing region. The image in Figure 10 shows detected lines and stabilized text. Because lines in man-made structures (buildings, walls, street lines, windows, etc.) are often longer and straighter than natural lines (branches, clouds, body parts, etc.), weighting longer lines more heavily in the angle calculation results in better stabilization. Text is then rotated to the resulting angle for each frame.

Algorithm:

```
STORE all detected lines from image in an array
FOR all lines in array
  CALCULATE line angle and length
  Compensate for the fact that detected horizontal lines are perpendicular to real world gravity
  UPDATE average weighted angle with current angle value with weight based on line length
ENDFOR
ROTATE text to average weighted angle of all lines
```

The device for which this algorithm is most useful is likely a head mounted display with a built-in camera. Since HMD input devices are typically small and constricted by low processing power, I also developed a way to use the above algorithms with sparse sampling. Instead of rating the region around every single pixel for darkness and uniformity, every *n*-th pixel can be sampled for similar results. For example, when a 250 x 200 pixel area for text is desired, sampling every 5th pixel for both *x* and *y* would still result in a sample size of 50 x 40 (2000 pixels per rectangle), more than enough to calculate ratings for darkness and uniformity. This means that the method can be scaled for smaller devices with limited capability.

3.4.6 Experimental Evaluation

In order to provide an initial evaluation of the method, I sought to determine whether the system will pick similar regions to a human when choosing locations for text and to see if those locations are preferred to fixed location configurations. To test this, I set up a pilot experiment where participants picked a region of an image which they thought was most appropriate for text. From each participant's selections, I could then compare how close their selections were to the selections of my system. I would also be able to compare whether this system was better at selecting a location for text than if left at a fixed position on the image, centered for example. This type of experiment provided a basic comparison of the algorithm's selections and human decisions.

3.4.6.1 **Experiment Setup**

The experiment included 19 participants, ranging from age 19 to 67, with an approximately even number of males and females. Participants were asked to view a set of 20 images with the same 25x25 matrix of bright green character strings (A1 to Y25) overlaid onto each image as shown in Figure 11. All images were 1024x768 pixels in resolution and viewed on participants' personal computer monitors. Spacing was set at 30 and 37 pixels between vertical and horizontal character strings, respectively. Images were chosen prior to applying the algorithm to avoid bias, so I did not know what region the system would select or what region users would pick in advance. I sent a web based questionnaire to each participant that contained all 20 images. They were then asked to enter the character string corresponding to the most appropriate region for text in each image, and took as much time as they wished.

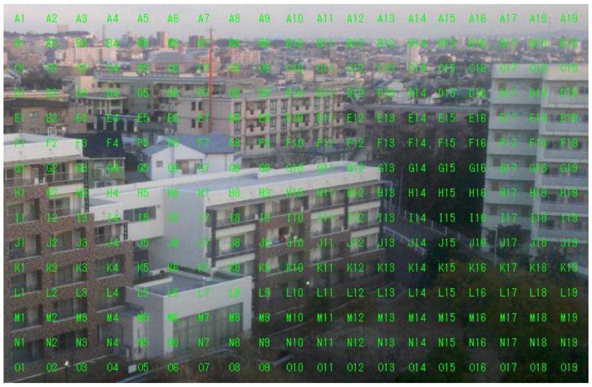

A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11 A12 A13 A14 A15 A16 A17 A18 A19
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12 E13 E14 E15 E16 E17 E18 E19
F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 F13 F14 F15 F16 F17 F18 F19
G13 G14 G15 G16 G17 G18 G19
H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11 H12 H13 H14 H15 H16 H17 H18 H19
J1 J2 J3 J4 J13 J14 J15 J16 J17 J18 J19
K1 K2 K3 K4 K5 K6 K7 K8 K9 K10 K11 K12 K13 K14 K15 K16 K17 K18 K19
L1 L2 L3 L4 L5 L6 L7 L8 L9 L10 L11 L12 L13 L14 L15 L16 L17 L18 L19
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M11 M12 M13 M14 M15 M16 M17 M18 M19
N1 N2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12 N13 N14 N15 N16 N17 N18 N19
O1 O2 O3 O4 O5 O6 O7 O8 O9 O10 O11 O12 O13 O14 O15 O16 O17 O18 O19

Figure 11 Segment of an image overlaid with green character strings showing the experiment format.

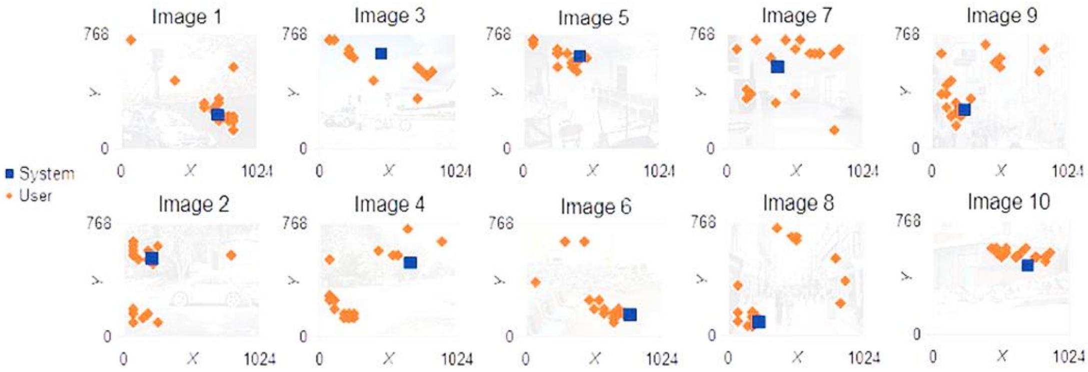


Figure 12 Plots of user selected points and system selected points for 6 of the 20 experiment images (semi-transparent).

Though conducting the experiment in various outdoor and indoor locations with both HMD and HUD systems would be more appropriate to evaluate usability, I first sought to conduct a simple comparison of human and system choices. Using images allowed us to test the algorithm independent of device and let us pick geographically dispersed locations as well as a large number of backgrounds. Though aspect ratios and color variations of monitors differ, the effects were considered negligible since all images were viewed on the same monitor.

3.4.6.2 **Results**

To show whether the system selected similar text regions to the human participants, I first plotted both user selected and system selected points for each image as shown in Figure 12. User selections are orange diamonds and system selections are blue squares. *X* and *Y* coordinates of each plot correspond to the *x* and *y* coordinates of each image in the experiment. Next, I calculated the Euclidean distance between all 20 user selected points and each system selected point for each image. For example, in Figure 12, Image 1 has an average Euclidean distance of 157 pixels, and Image 4 has a distance of 498 pixels.

Table 1 Average Euclidean distances between all 20 user selected and system selected points for each experiment image.

Image # (bold columns) and average Euclidean distance (non-bold columns).							
1	157	**6**	231	**11**	401	**16**	338
2	211	**7**	302	**12**	369	**17**	240
3	324	**8**	303	**13**	129	**18**	253
4	498	**9**	279	**14**	210	**19**	153
5	170	**10**	160	**15**	183	**20**	432

The first 10 plots are shown, which should provide a good visual comparison of user and system selected points. The average Euclidean distances for each image are shown in Table 1, and the average for all 20 images is 267 pixels. With these results, I cannot state that the method completely resembles human behavior, however, through qualitative analysis I can make the claim that my system chooses similar regions to a human to a certain degree. For example, in the plot of Image 1 in Figure 12, it is clear that my system picked a point similar to a majority of user selections. 16 of 19 users chose to place text in the lower right corner of the screen, just like the system did. 10 of those 16 people chose a location within 100 pixels of the system selected location. 6 of those 16 people chose a location within 160 pixels of the system selected location. The remaining 3 people selected a point on a different section of the image that was over 300 pixels away from the system selected location. So, for all images, I show how many user selected points were within 50, 100, 150 and 200 pixel distances of the system selected points from all images, which are listed in Table 2.

Points within 50 pixels would be extremely close, whereas over 200 pixels would not be considered close or in the same area. For reference, on a standard 17 inch monitor, 50 diagonal pixels are equivalent to about 0.66 inches (1.5 centimeters). I then compared my system selections to 5 different fixed view configurations, including upper left (UL), upper right (UR), lower right (LR), lower left (LL), and center (C).

Center text was located at the very center of each image, *y* positions of upper and lower texts were at 330 pixels above and below center, and *x* positions of left and right texts were at 250 pixels left and right of center. As a quantitative comparison, I showed the average Euclidean distance between all configurations and user selected points, and conducted a one way analysis of variance (ANOVA) for the system vs. each fixed view configuration as shown in Table 3. Images were established as the control, and system, UL, UR, LR, LL, and C were independent variables, allowing for a pairwise comparison of my system to each fixed view configuration. The results of F show a strong effect between Euclidean distances of system selected points and all fixed configuration locations.

Table 2 Number and percentage of points selected by the user within various Euclidean distances from system selected points.

Euclidean distance to system selected points in pixels	**Number of user selected points within that distance (out of 380 points total)**
200 pixels	187 points (49.2%)
150 pixels	149 points (39.2%)
100 pixels	105 points (27.6%)
50 pixels	34 points (8.9%)

Since the averages of system selected locations were closer to any of the fixed selections and since an effect was shown for each pairwise comparison, I can conclude that locations selected by my system are preferred to locations permanently fixed to points on the presented images.

One important point to note is that several images had several good (though not ideal) points that were selected by users. For example, a majority of user selections in Figure 12, Image 2, are grouped in two areas, the upper left and lower left side of the image. My algorithm picked the location nearest to the majority of user selections in the upper left corner; however, this is not reflected when taking the average Euclidean distance between all 20 user points and the system selected point. Secondly, though the experiment I conducted is adequate to show the basic effectiveness of my algorithm, an experiment using see-through optical HMD and or HUD systems would be necessary to fully measure usability.

3.4.7 Dealing with Scene Complexity

While the suggested methods provide a basic framework for moving and stabilizing an overlaid text throughout the real world, they are by no means comprehensive. However, I believe problems with more complex environments can be solved with feature tracking, object detection, or other context analysis algorithms. Especially when attempting to manage user-centric text, my methods provide a good foundation. Although the algorithm deals mostly with the viewing of text, users still need methods for interaction. At the very least, a user should be able to scroll through text and transition between windows or programs. Though camera tracking of finger and hand are possible for gesture recognition, a method that is likely more simple and intuitive is to give the user a wireless touch based surface to interact with. Using a touch-screen phone, belt-worn touch pad, or touch screen watch will allow click, scroll, and relative pointing and would not require constant visual contact.

Table 3 Table showing average Euclidean distances between each configuration and participant selected points across all images (left) and pairwise ANOVA results between system and fixed configurations (right).

Average of Euclidean distances between all user selected points in all 20 images and:		**ANOVA for system vs. fixed configurations**	
System Selected	267 pixels	**$F_{1,18}$**	**P**
LR Fixed	557 pixels	50.6	< .01
LL Fixed	477 pixels	23.5	< .01
UL Fixed	423 pixels	15.5	< .01
UR Fixed	511 pixels	55.1	< .01
C Fixed	342 pixels	12.4	< .01

In addition to scrolling through text or window management, there are also a variety of input methods that not only allow users to enter text while walking but do not require the user to maintain constant eye contact with the input device. Devices such as the AlphaGrip, Torso Keyboard, Gestyboard, Twiddler, and voice recognition do not need constant visual contact like a software keyboard would (Coskun et al., 2012, Lyons et al., 2004).

As scene/lighting complexity increase, more specific methods for dealing with variation become necessary. In addition to the software solutions I mentioned before like feature tracking, there are several methods appropriate for dealing with some of these complexity challenges. For example, in noisy images where there may be many small viable regions instead of several large ones, text can be scaled down in size or broken up into smaller bits. Twitter feeds or text messages could be placed throughout the environment in a logical manner much like signs are placed on either side of a street.

Lastly, for wide field of view applications, a fish-eye lens could be affixed to the camera to increase the area of analysis. In this case, instead of moving text to new regions on the HMD when previous regions leave the viewing screen, an indicator could be used to show the direction of off-screen text or other content, especially for mobile workspaces (Billinghurst et al., 1998). Messages, feeds, and miscellaneous smaller notifications would be ideal for this type of display.

3.4.8 Algorithm Optimization

Though thresholding text movement from one best position to the next results in some stabilization, a higher degree of stability is needed for users to be able to view text consistently. Part of the reason that text moves unpredictably is that a many best points may exist if a large and uniform surface is present in the image. To solve this problem, I implemented a combination of a weighted centroid calculation and level-setting to determine a stable best point. This can be represented by $C_{(x,y)}$ below, where *n* is the total number of good points, *V* is the viability rating of a single point, and *x and y* are the coordinates of each good point. Each point is weighted based on the sum of all surrounding viability ratings, represented by $\sum V$.

$$C_{(x,y)} = \sum_{i}^{n} \frac{V * (x,y)}{\sum V}$$

In short, the coordinates of the centroid are calculated using weighted values of the best point as well as other relatively good points in the image. This provides much more stable placement since the average location of many good points in the image is less likely to change quickly than a single best point. However, if two or more good regions in the image exist, taking the centroid of those regions may result in text being placed between multiple regions, and in a less visible location. To prevent this from happening, level-setting is used to eliminate the smaller of the two regions.

3.4.9 Experiment with Real Time Videos

Many prior studies have been limited in scope and are not designed for use with mobile applications. Considering the limited results concerning text readability, I found that a broader user study is required to describe how individuals would choose to overlay content onto their immediate environment given free choice, especially in real time. Next, I describe the results of experiments designed to compare the selections of the automatic system with those of humans on real time videos taken from a pedestrian's point of view. In addition to comparing human and system selections, I conduct a thorough analysis of placement tendencies in order to learn more about how HMD users would overlay text.

In order to further compare my algorithm to human tendencies and to learn more about how users think about text overlay, this experiment was designed to study what happens when users are given free control of overlaid text in real time. More specifically, I answer the questions: 1) How does the algorithm compare to text overlaid in a real time, dynamic environment? 2) Where and in what way do users tend to place text? The results of this experiment are useful for development of future view management algorithms and displays, and also provide insight into how people think about text overlay.

This lets us observe on a frame by frame basis how users would manually place text when looking at the world through an HMD. A total of 20 people from ages 18 to 32 participated, with an approximately even number of males and females. Prior to the experiment, four videos of different locations on a college campus were taken from a high definition (HD) 1080p video camera held at eye-level. The cameraman traversed different locations including the inside of a building, walkways, areas with a large amount of foliage, and more open meeting areas to present a variety of different situations to users. All videos were taken at mid-afternoon, where the sun and lighting conditions would often interfere with text readability. Though having users place text in the real world outdoors would be ideal, controlling outdoor lighting conditions and ensuring all participants' head movements were the same would be next to impossible. Therefore, I opted to show videos using an HMD, which ensured that the environment was viewed in the same way each time, and prevented user head motion from affecting perspective.

3.4.9.1 **Participant Tasks and Data Capture**

The tasks for all participants were to use a mouse to overlay text onto the videos in real time. Each participant sat at a desk and was asked to wear a Silicon Micro ST1080 HD HMD (45 degree field of view, 1920x1080 pixels). Participants with glasses were allowed to hold the display with their left hand for stability during the experiment, and all participants confirmed that they could see text on the screen before starting. Using a mouse, participants were instructed to overlay text onto each video in the most visible and appropriate location on the screen. No clicking was necessary since simply moving the mouse recorded each (x, y) coordinate position at approximately every 10 milliseconds (ms). Though three participants discontinued the experiment due to motion sickness, another 20 completed the experiment to the end.

As controls, all participants viewed the same four videos, all of which were three minutes long. Viewing order was randomized and counterbalanced between participants. Text was limited to bright green, which was chosen for its visibility and contrast with most man-made objects (Gabbard et al., 2006). Three different paragraph sizes and transparency levels were presented in order to see whether they would affect the frequency at which participants moved text to new locations. To conduct an analysis of all 20 users at the same time, I needed to perform a side-by-side comparison of cursor positions in real time, so all mouse movements and videos were synchronized.

3.4.9.2 **Results**

In addition to providing a comparison of the algorithm's placement to the dynamic placement of text by users in real time, I present several other useful findings. First, users tend to have a general tendency to place text just below screen center. Second, frequent targets for overlay tend to have similar characteristics and can be classified. Last, differences exist in placement frequency for varied text transparency, but not paragraph size.

The comparison in this experiment was conducted by manually selecting 12 frames from all 4 videos which exhibited the most tightly grouped user selections, representing common agreement for text placement. Row B in Figure 13 shows user selected points (black crosses) from 3 original video frames, providing a visual representation of a cluster of common selections. Instead of comparing the distance between user points and the system's best point as in the first experiment, the comparison this time shows how many user selected points are within the algorithmically selected viable region shown in Figure 13, row C.

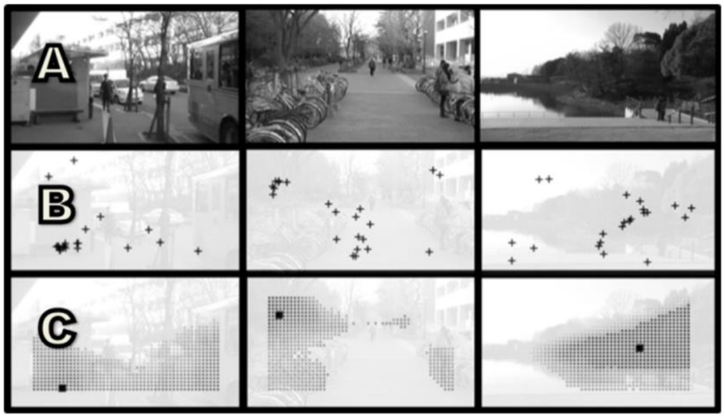


Figure 13 Sample images showing original images, user placement, and system placement (in greyscale for simplicity). Row A) Three representative frames from experimental videos shown real time. Row B) shows corresponding text placement by users. Row C) shows the best points selected by the algorithm (large dark squares) and viability (darker = more viable).

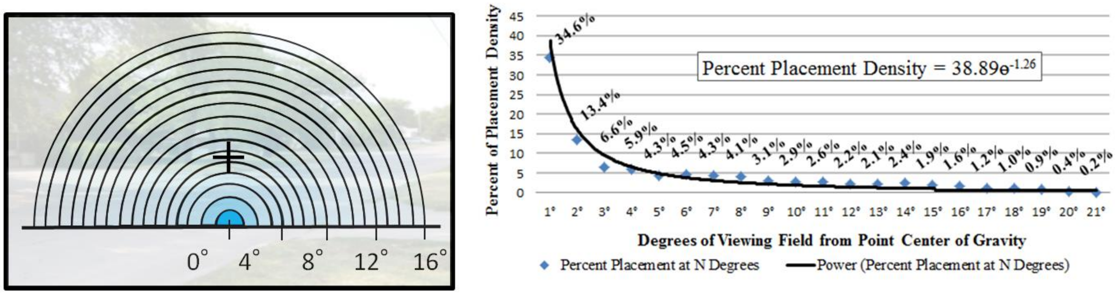


Figure 14 Positioning density of text across all videos, with the gravitational center of placement at 0 degree, representing 39% density, versus 1.6% density at 16 degree (left). A cross representing screen-center is overlaid onto a semi-transparent scene image for reference. The plot showing the power relationship for placement density is also shown (right).

Out of 240 user selected points, 161 fell inside of the region selected by my algorithm, an overall 67.1% hit rate. In 8 of the 12 video frames, the hit rate exceeded 75%. The failure of my algorithm in the remaining 4 frames (hit rates of 65% or below) was often due to the fact that a very dark, visible location was available in a corner of the screen, but was far away from the central region of the viewing screen. To determine whether users had an overall tendency for placement near screen center, I next conducted a longitudinal analysis of placement.

In order to show where in the HMD viewing field users were placing text over time, I plotted all user data over the span of all videos and calculated point density for each degree of field of view from screen center as shown on the left of Figure 14. Each arc represents a degree of field of view, and the darker the hue, the higher the density. The image covers a 42 by 23.6 degree field of view from the camera's perspective.

The horizontal center of gravity for all points, located at the center of the arcs, rests at a point 5.6 degrees below screen center (represented by a dark cross). I found that this density distribution can be modeled by the power relationship shown below, where D_θ represents placement density and θ represents the degree of field of view.

$$D_\theta = 38.39 * \theta^{-1.26}$$

D_θ can be used to calculate the probability that a user will place text in a certain degree of field of view and to eliminate some of the observed failure cases.

In addition to identifying the 12 video frames with the most evident clusters, I sought to utilize frames from all videos in order to classify general targets for text placement. To initially identify potential clusters, I calculated the Euclidean distance between each individual point in a frame and the center of gravity of all points. The lower the average distance was in a frame, the tighter the grouping of user placement. Local minima were taken to represent clustering, and a visual confirmation was also conducted, resulting in the clusters shown in Table 4. The column labeled "5-10 users" represents a 3 second timeframe where between 5 and 10 users continuously overlaid text onto the same surface or onto an area that exhibited similar characteristics, such as shade. The next column is the same, but for cases with more than 10 users. A higher number of users in the cluster shows when placement tendencies had a higher affinity for a certain surface or characteristic. The “Total” column represents the total number of times a single user was within either of the two aforementioned cluster categories. Characteristics were logically picked based on previous readability research and on targeted features of current automated algorithms such as uniformity and feature quantity (Chen, 2004, Orlosky, 2013).

Regarding uniformity classifications, 'high' is defined as a surface or area where both lighting and color are uniform, 'medium' as uniform lighting and contrasting color or uniform color and contrasting lighting, and 'low' as both contrasting lighting and color conditions. Across videos, paragraph size and transparency were also varied to determine whether there was effect on movement speed. Higher transparency was included to simulate brighter lighting conditions. If this forced users to move text to new locations more quickly, it would make more sense to decrease the movement threshold. Similarly, users might try to find better spaces for larger blocks of text.

I found that there is a significant difference in movement frequency for the transparency condition, but not for text size, as shown in Figure 15. The two bars for videos 1 and 2 show movement with and without transparency, and a one way ANOVA shows a strong effect (Video 1: $F_{(1,19)}$=56.8, P<.01, Video 2: $F_{(1,19)}$=89.3, P<.01) of transparency. Unexpectedly, there was little statistical significance in variations of paragraph size (Video 3: $F_{(1,19)}$=3.95, P=.047, Video 4: $F_{(1,19)}$=0.21, P=0.64). I suspect this may be because users only focus on a single line of text at any given time regardless of paragraph size.

Table 4 Classification and frequency of common surfaces and characteristics. The second and third columns represent the number of times a cluster was observed for that item. The last column represents the total number of individual placements.

Surface	**5-10 users**	**10+ users**	**Total**
Wall	8	10	140+
Road/walkway	7	10	135+
Foliage	12	5	110+
Other	11	1	65+
Characteristic			
Shaded	31	21	365+
Grey Variant	23	17	285+
Vertical Surface	23	15	265+
Medium Uniformity	24	14	260+
Horizontal Surface	13	7	135+
High Uniformity	6	9	120

3.4.10 Discussion

From this series of experiments, we can learn a great deal about how HMD users think about text overlay. Though there is a common misconception that HMD technology is distracting and can lead to accidents, through the pilot experiments, I find that in comparison with smartphones, HMDs may actually provide users with increased contextual awareness without a significant reduction in performance. It should be noted that this comparison is between an HMD and smartphone, but no baseline without a device was taken. Still, these results may help debunk some of the myths regarding awareness in new wearable displays. Additionally, the results show that using an HMD may be safer than looking down at a cell phone while walking or cycling.

Though users are provided with increased awareness of their surroundings, they also experience the unwanted effect of merging the user's immediate field of view with a virtual field of view. To solve this problem, automatic text placement, especially for user centric information such as e-mail, is a potential solution. As evident in the third and fourth experiments comparing user preference of text placement with my algorithm, intelligent selection of areas that maximize viewability can be used as a method to mimic human text placement behavior. By examining common groupings of placement, it becomes easier to infer what types of mental models are being used for text placement. Though algorithmic placement resembles human choices to some degree, the method is somewhat limited in that it is not tested for use in a variety of different scenarios. Further testing is necessary for different geographic, spatial, and interpersonal settings to ensure general usability.

3.5 Decreasing Invasiveness and Managing Interpersonal Communication

Although Dynamic Text Management provides a good way to improve the visibility of virtual content, it can still interfere with day to day activities, especially if it resides in the user's central vision. In particular, messages, notifications, or navigation instructions overlaid in the

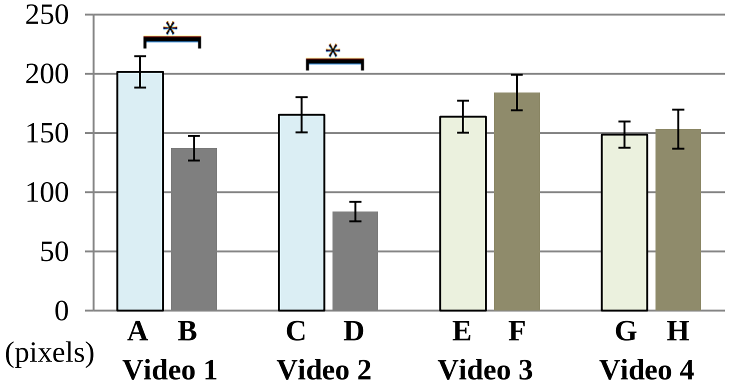


Figure 15 Average movement of all users in a particular video by condition (in pixels per second). Independent variables are 80% transparency for A) and C) vs. no transparency for B) and D) and small text block for E) and G) vs. large text block for F) and H).

central FOV may then become a barrier to effective face-to-face meetings and everyday conversation. Many other text and view management methods attempt to improve text viewability, but also fail to provide a non-invasive personal experience for the user.

Next, I introduce Halo Content, a method that proactively manages movement of multiple elements such as e-mails, texts, and notifications to make sure they do not interfere with interpersonal interactions. Through a unique combination of face detection, integrated layouts, and automated content movement, virtual elements are actively moved so that they do not occlude conversation partners' faces or gestures. Unlike other methods that often require tracking or prior knowledge of the scene, this approach can deal with multiple conversation partners in unknown, dynamic situations. In a short experiment with 14 participants, I show that the Halo Content algorithm results in a 54.8% reduction in the number of times content interfered with conversations compared to standard layouts.

3.5.1 Introduction

With the growing number of wearable, head worn, and head up displays, the need to manage content in a user's field of view is becoming increasingly important. Products like the Google Glass and Epson Moverio give users the ability to overlay virtual information directly onto their field of view, allowing for improved information display while mobile. There have accordingly been many attempts to address related view management problems, many of which focus on improving content readability and visibility (Orlosky, 2013). Additionally, many algorithms have been designed to effectively manage labels on environmental objects and the resulting virtual clutter from those objects (Bell, 2001, Hartmann, 2004, Makita, 2009). Other management schemes tend to focus on occlusion problems and making sure both content and labeled object are consistently visible (Ajanki, 2011, Zhang, 2005).

However, these methods typically focus on environment-centric text, which refers to labels that are previously registered to existing objects in the real world or in pre-defined content (Orlosky, 2013). In contrast, user-centric items such as e-mails or personal notifications have only recently been targeted for mobile view management. Unlike labeling of known 3D objects or environments which may be stationary, view management of user-centric information must often rely on real time analysis of a more dynamic environment.

Here, I focus on content that can interfere with interpersonal interactions. For example, a pedestrian reading notifications or following navigation instructions in a head worn display may stop to ask for directions. My goal is to prevent virtual information from interfering with the following conversation or interpersonal interaction. To accomplish this, the algorithm detects faces in the scene and moves content along a series of layout dependent vectors, pushing it up and away from its usual fixed screen location.

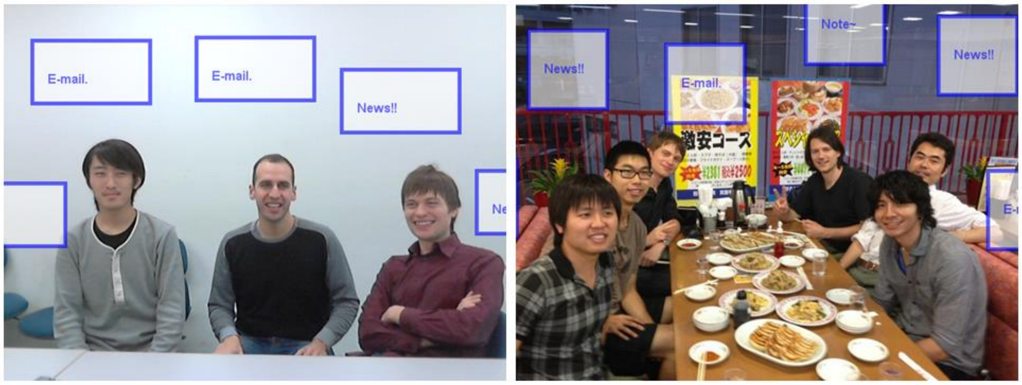


Figure 16 The Halo Content algorithm applied to billboard style text notifications in conversations with multiple participants in different environments.

This prevents text from occluding other people in the field of view, as shown in both images in Figure 16. This strategy can be applied to various social situations, chance outdoor meetings, and everyday conversations. More specifically, I first utilize face detection to search for potential interaction targets in the environment. Faces are then constantly evaluated for persistence (whether or not the detected face is still in a user's field of view despite detection failures), and for conversation potential (the probability that a persistent face will engage in conversation at a certain distance). Once the face analysis portion of the system is complete, a layout management algorithm then actively moves content to ensure that other people in the user's field of view remain visible.

In many cases, the algorithm forms what looks like a halo of content around other people in the environment, as can be seen in Figure 17 C and Figure 18 B, hence the name Halo Content. In contrast with other similar algorithms, my system can deal with numerous environmental objects, handles multiple conversation partners and screen elements, and allows for temporary off-screen placement.

3.5.2 Prior Work

Up to now, many view management algorithms have been proposed to manage virtual content. A majority of related algorithms attempt to minimize occlusion of virtual labels relative to a target object. For example, Tatzgern et al. define 2D and 3D labeling techniques to ensure that both labels and leader lines do not cross or occlude each other (Tatzgern et al., 2014). Similarly, Reitmayr et al. propose a method for semi-automatic annotation in combination with simultaneous localization and mapping algorithms (Reitmayr et al., 2007). Makita et al. affixed trackers to users in the real world, and developed a method for managing annotations around the users’ bodies as they moved along in real time (Makita et al., 2009). Most of these strategies are good for virtual immersion applications, gaming, when labels are fixed to a single, stationary location, or when 3D knowledge of the scene is already known. However, they are not necessarily ideal for mobile environments or face-to-face conversations. Several other strategies for managing mobile content include physical interaction strategies, for example pasting content on a nearby surface (Ens et al., 2014). Other more specific automation strategies have also been applied for managing content in vehicles, such as that of Tsai et al. (Tsai et al., 2013).

Though Dynamic Text Management uses background color and texture to maximize viewability, object recognition was one potential way to deal with other environmental situations, which has been implemented in Halo Content. In contrast to other works, Halo Content focuses on the user’s interpersonal interactions, rather than readability or virtual clutter. Simply put, the strategy is to combine object recognition techniques with layout management to achieve augmented reality that is non-invasive. Additionally, no prior knowledge of the scene is necessary, both multiple conversation partners and virtual elements can be dealt with, and the approach can be applied with other object recognition algorithms for real-time use.

3.5.3 Framework and Layout Methodology

First and foremost, I wanted the user to be able to carry out everyday conversations and activities without having to worry about closing and/or managing text. To accomplish this, three primary steps are used to prevent text from entering the conversation/gesture area. These steps include defining layouts, face detection and processing, and managing direction and movement of screen items based on position, size, and number of detected faces.

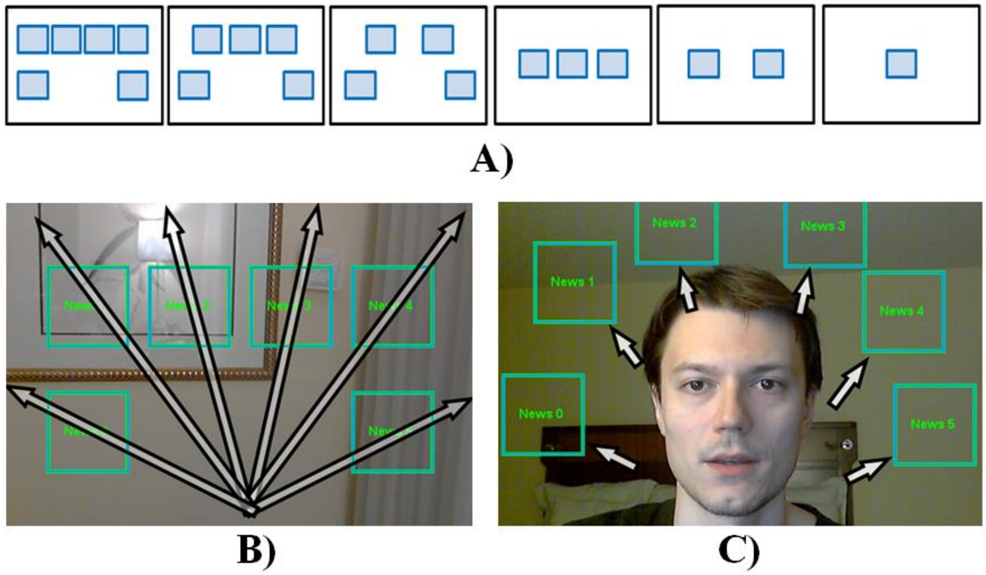


Figure 17 Diagram of content layouts allowing for between one and six virtual elements, B) example of movement vectors for a six element layout, and C) content that no longer occludes the conversation.

3.5.3.1 **Defining Layouts**

Although environment centric layout management methods can usually manage multiple elements (Bell, 2001, Tatzgern, 2014), current user-centric text management systems often focus on the current window of content in the user’s field of view (Orlosky, 2013). Since users are often presented with a number of different user centric data items or notifications with limited screen size, I sought to manage multiple items in a constrained space. I chose layouts in which content can simultaneously move away from objects of concentration, and still have a minimal chance of occluding other content, regardless of the size and number of faces in the user's field of view.

Although a number of different layouts are possible, I predefined layouts for anywhere between 1 and 6 blocks of content for demonstration and testing, as shown in A of Figure 17. Each of these layouts is designed in such a way that if any element is moved outward, it has a minimal chance of occluding an adjacent element. Additionally, order is always preserved during movement. This means content will always appear in the order the user last left it, preventing users from having to search for icons or widgets that have moved to a different screen location. Addition or removal of elements can also be accomplished without reordering. A simple example of how content would move within a layout is shown in B and C of Figure 17. Virtual elements lie on a number of vectors that run through the center of each piece of content. Each element is then checked to see if it occludes any faces in the scene, and moved away from its original location by a user-defined distance to prevent occlusion.

3.5.3.2 **Face Detection and Persistence**

The face processing library I used is from OpenCV, which employs a Haar classifier for face detection. Alone, this is not enough to guarantee consistent detection and smooth, consistent layout management. The problem of inconsistent detection, also referred to as a persistence problem, exists with many real time detection algorithms, including detection of markers for augmented reality and optical character recognition. In this case, regardless of several failed detection frames, content should still remain out of the path of the user's conversation. In order to accomplish this, we first define a persistence variable (*Pf*) for each detected face, which functions like a threshold. Once a face is detected, it is loaded into a resizable array with a predefined *Pf*, the detected size, and *x*,*y* position. If multiple faces are detected, they are all inserted into the array within the same frame.

After all detected faces have been processed, the persistence variable of any detected faces that had previously existed within the array is decremented by one. Any instance of a detection that has persistence of 0 or below is removed from the array. As a result, faces that have existed in at least one of the last *Pf* frames affect the content layout algorithms below. This means that even if face detection fails in several frames, content is still kept away from the person or people in the conversation. *Pf* regulates how long a face persists within the array, so a higher *Pf* (assuming more detection failures) will keep content away longer. One other benefit of this approach is that text movement exhibits a smoothing effect since faces are more consistently present.

3.5.3.3 **Direction, Movement, and Dealing with Multiple Faces**

In contrast with text readability, I am more concerned with the viewability of people in the conversation. Therefore, I designed a view management method that prioritizes visibility, is less invasive, and provides easy access to off-screen information. Using the previously mentioned layouts shown in Figure 17, vectors are first defined that start from a point at the bottom center of the screen (x_{bc}, y_{bc}) and run through the centroid of each virtual element at angle θ, as shown in images A and C of Figure 18. For multiple elements, the vectors run outwards towards the left, upper, and right borders of the screen as shown in C. Content can then move along each of these vectors whenever a face comes too close to a virtual element. Movement logic on the vector and distance (*Lc*) from (x_{bc}, y_{bc}) can be described by:

$$\begin{array}{c} IF(Dv < thresh \\ \&\& \\ Fbc < Cbc) \end{array}, then\colon\ Lc = \ dmin + (Fbc - Cbc)$$

where *Dv* is the minimum distance from the detected face to the nearest vector, *dmin* is the desired minimum distance from a detected face to moved content, and *Fbc* and *Cbc* are the Euclidean distances from each face to the origin and the content block under consideration to the origin, respectively. Each element is then moved along its respective vector so that there is a final distance of *dmin* pixels between the element and the nearest face. As seen in C and D of Figure 18, multiple faces are handled in a similar way to a single face. For every virtual element present, the distance to each face in the scene is first checked, and the element is then moved if necessary. For example, virtual elements in Figure 18 C (blue boxes) are moved dmin away from detected faces (green boxes). When a virtual element occludes more than one detected face, the closer face is used in the calculation.

This idea was inspired by previous strategies that employ potential fields for content movement (Hartmann, 2004). In the case of multiple faces occluding a single element, movement is based on the nearest face with respect to the origin (X_{bc}, X_{bc}). Since processing power is a concern for mobile devices, the distance is measured from a bounding box on both rendered content and nearby faces, thus simplifying the calculation, much like Minkowski Sums are used to detect collisions in gaming applications (Van Den Bergen, 2001). A running average is also taken for the position of each element, so content appears to smoothly move away from any faces coming into the user's field of view. A representative sample of resulting movement for different element layouts and number of conversation partners is shown in B and D of Figure 18.

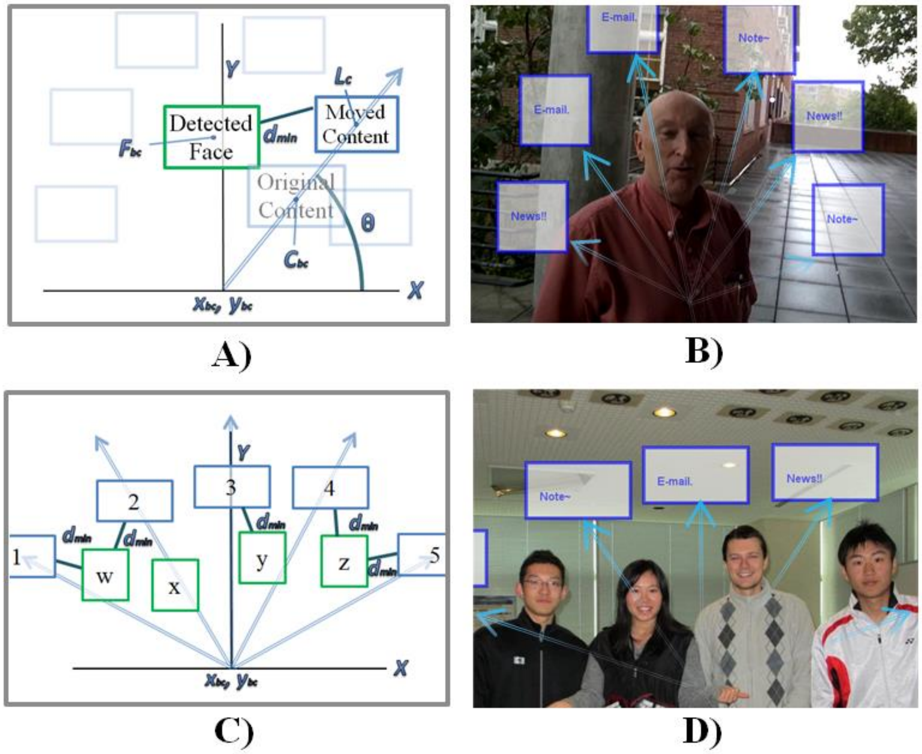


Figure 18 A) Diagram showing the geometry of displacement direction and distance calculation for an individual vector, B) corresponding managed content for one person and 6 virtual elements, C) geometry for a multi-face example with 4 people and 5 elements, and D) corresponding managed content. (Note that two elements in D are off-screen.)

Algorithmically, this means that the system loops through the array containing persistent faces, and checks for nearby virtual elements in every frame. This logic is defined by the pseudo code shown below:

```
WHILE camera is on
   RUN face detection on current frame
       ADD any detected faces to persistence array with $Pf$
       DECREMENT $Pf$ of any existing faces in array by 1
           FOR each virtual element
              FOR each face in persistence array
       IF distance between face and content vector ($D_v$) of nearest element < $d_{min}$
            AND face-origin distance ($F_{bc}$) < content-origin distance ($C_{bc}$)
       THEN use $d_{min}$ to set new content distance ($L_c$)
                ENDFOR //incremented through all faces
           ENDFOR //incremented through all virtual elements
ENDWHILE
```

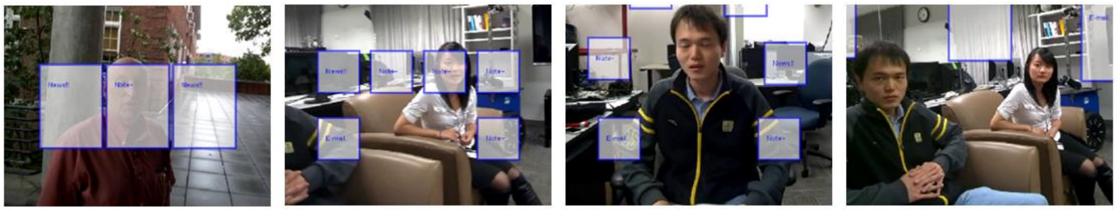


Figure 19 Four of the images used in the experiment. The left two images show a default layout, and the right two images show content managed by the Halo Content algorithm.

Although less likely, the case of a user's face entering from the top of the screen must be dealt with differently since content blocks are never migrated downwards to avoid occluding bodies or hand gestures. In this case, as soon as a new face comes within d_{min} of the vector corresponding to the closest element, the element is migrated up and over the face and kept at d_{min} pixels away from the face from that time forward. Though I am still testing for other exceptions, the movement scheme I propose appears to generally solve occlusion problems in this context.

3.5.4 Evaluation

3.5.4.1 **Setup**

When evaluating the system, I conducted a simple test to find out how well the Halo Content algorithm can prevent text from interfering with a number of different conversations. To do so, I asked participants to view a variety of frames taken from 3 different videos. Simulated content was then managed for each frame with the Halo Content algorithm, and compared to corresponding standard layouts as baselines. Participants evaluated content on each frame as “would interfere” or “would not interfere.” Several images are shown in Figure 19 for reference. I started by taking three videos from an HMD camera encompassing a variety of different conversational situations, including a chance outdoor meeting, a four-person research discussion, and an in-office consultation. These videos were taken from a first person perspective using a head worn display so that head movements and interactions would be recorded. 20 frames were extracted randomly from each of these videos, and frames without faces were rejected and replaced at random. I then applied the Halo Content algorithm to 3 different standard layouts, and for combinations of 2, 3, and 6 simulated blocks of content as shown in Figure 17 A, for a total of 60 processed frames at 640x480pixels (px). I also created a corresponding set of 60 frames with the same 3 layouts and block sizes, but did not apply the management algorithm to provide a baseline for comparison.

The blocks of content were displayed as white, semi-transparent billboards containing a single randomly selected text notification. Sizes differed with respect to number of blocks present, with 200x180px, 150x200px, and 100x100px for the 2, 3, and 6 block layouts, respectively. A total of 14 volunteers, 9 male and 5 female, with a mean age of 31.9, participated in the experiment. I employed a within subjects design, where each participant evaluated the management method on each of the 120 frames, for a total of 1680 evaluations. The order of conditions in each interface was randomized between participants to alleviate ordering effects.

3.5.4.2 **Results**

Across all participants (group A), 86.5% of frames using layouts without management were evaluated as interfering, in comparison with 31.8% for those using Halo Content. A two-way analysis of variance shows a significant difference ($F_{(5,13)}=26.42$, P<.0001) between ratings of the two display methods across all sizes and numbers. Percentage of content rated as interfering with respect to management method and number of elements is shown in A of Figure 20. I also noticed a clear division of ratings within the experiment. Out of the 14 participants, 5 rated a majority of content as interfering, regardless of whether it was managed by Halo Content or in a static layout. For the remaining 9 (group B), interference of the static layouts remained about the same, but resulting interference of Halo Content was significantly reduced. As shown in B of Figure 20, only 11.3% of frames managed by Halo Content were evaluated as interfering for these participants, resulting in a 73.3% reduction in interference compared to static layouts. A slight effect was found for number of elements ($F_{(2,8)}=2.67$, P<.05) for this group, which suggests that an increasing number of elements may result in increased interference.

The initial results of the experiments suggest that Halo Content is a good way to prevent certain types of augmentative and virtual content from becoming invasive in conversational situations. As HWDs, AR applications, and virtual content increase, it will make sense to have a number of management algorithms in place for different situations. For example, Halo Content might be used for conversations, whereas 3D labeling techniques or visibility management might be used for industrial tasks. Of course, there are tradeoffs between using this vector based strategy versus other algorithms. For example, it may be difficult to mix environment relative labeling with user-centric e-mails.

3.5.4.3 **Discussion**

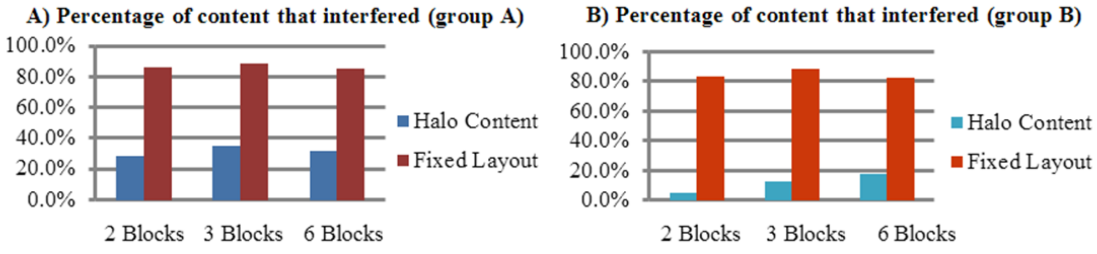


Figure 20 A) Table showing interference of the Halo Content algorithm compared to typical on-screen layouts for 2, 3, and 6 element layouts. B) The same as A, but excluding the 5 users who were unsatisfied when almost any virtual content was present in the scene.

While vectors provide a very fast way for a user to manage mobile content, adaptations would also be necessary to view content with a shared view or perspective. Although to some extent I provide robustness to failed face detections in this algorithm, it is important to note that this does not solve persistence problems completely. Additionally, placement depth needs to be taken into account. While the strategy employed in Halo Content works well for monoscopic HWDs with a fixed focal plane, stereoscopic displays would benefit from content placed at the same depth as the user's gaze (Tan, 2003). Eye tracking may be a potential solution to this problem. One other benefit of Halo Content is that it is applicable to devices other than see-through HWDs. There are many applications for immersive virtual reality, heads-up systems, and other static see-through displays. The algorithm and layout methods proposed here are easily adaptable to other virtual spaces. One good application example would be the migration of text or content away from vehicles for drivers when on the highway. The face recognition algorithm would be replaced with vehicle recognition, and layouts could be expanded for HUD sized displays.

3.6 Summary

This chapter describes a number of ways to improve the visibility of both content and environment. The first of these is called Dynamic Text Management, which analyzes background information and finds dark, uniform points on a user's path to maximize readability. The second of these methods is Halo Content, which utilizes face detection to move content away from other individuals in a conversation or interaction. This accounts for locations in the environment that may be dark or uniform, but still have a role to play in an individual's interpersonal interactions.

CHAPTER 4

Attentive Interfaces

Though content placement algorithms are well suited for managing visibility or readability of content, they often do not take user safety into account for mobile AR. The Halo Content algorithm described in the previous chapter may be able to accomplish this in certain situations such as pedestrian safety since a vehicle detection algorithm would work instead of face detection, but I realized that a more consistent method would be necessary to make sure content does become a hazard for the user. The term "attentive interface" has been used in the past to describe systems that attend to and prioritize information for the user (Maglio et al., 2000, Vertegaal 2002), which are well suited to the needs of HMD users. Therefore, I set out to construct an attentive interface that prioritizes user safety by quickly providing a clear view of the user's immediate surroundings when necessary. This chapter will first introduce the concept behind the system and prior work, followed by a detailed description of design and logic choices, and concluded with a series of experiments testing the feasibility and accuracy of the attentive interface.

4.1 Introduction

Virtual information can now be constantly displayed in a user's field of view, which can cause distractions and require users to interact in ways that may be tiring or unnatural (Orlosky, 2013, Woods, 2003). For example, text displayed on a sidewalk in front of a user should only be visible when the user wants to read it. Examples of text overlaid onto a potentially dangerous location can be seen at the bottom of Figure 1, as shown previously. If the user glances down at his or her watch or looks out for oncoming traffic, virtual text should be removed from his or her field of view as quickly as possible to prevent interference. Users typically need to press a button or perform some sort of physical action on the device in order to close or manipulate content. Not only does this take time, but it may be distracting and dangerous, especially in mobile situations. An appropriately designed attentive interface should reduce or completely eliminate this manual interaction in order to improve mobility and safety.

As one solution to this problem, I propose a combination of eye tracking with a multi-focal plane HMD. Currently, there is a lack of methods developed to improve the safety of monoscopic HMDs, which currently dominate the wearable display market. By taking advantage of users' natural tendencies to focus on objects of attention at different depths, the need for physical button presses or other manual interaction can be reduced. Though eye gaze has often been proposed as a form of interaction, most gaze based methods only show the direction a user is looking, but not whether the user is focusing on a more distant object in the same line of sight, for example a real car versus a virtual e-mail. This is where focal depth becomes very useful, since it can determine whether the user is looking at content on the display or at a hazardous object in his or her environment. This depth can then be used to automatically dim or close distracting content. In addition to automatic content dimming or closing, once a user looks back into the display, he or she should be able to quickly continue viewing content uninhibited. Instead of having to find and press a button or remove a touch-screen device from one's pocket, focus can again be used to re-engage an active window.

Users then have a more intuitive and robust interface for interacting with virtual content. Recent developments in display technology show that multi-focal plane displays will soon be commercially available, making this interface relevant to both current and future HMD systems (Urey, 2011). To some extent, this sort of interaction has been previously tested with stereoscopic 3D displays that sit in a static position away from the user's face (Kim, 2011, Kwon, 2006). To expand this research into the mobile domain, I use a glasses-type eye tracking interface, and combine it with a prototype HMD containing focal planes in the near, mid and far-field (approximately 30cm, 1m, and 2m+, respectively). Using this setup, 14 individuals were asked to participate in an experiment testing focus based interaction, and measured the variance of their eye convergence at each focal depth. Experiments show that convergence for nearly all users can be used to accurately select objects on any of the 3 focal planes, and that certain depth cues do not have an effect on the eye's physical focal tendencies.

4.2 Prior Work

Since the advent of the head mounted display, accurate image reproduction has been a goal of head mounted display research. Accommodation in particular is difficult to reproduce since the focal depth of an HMD must be at a variable distance depending on the eye's current focal point. Though not a wearable device, one of the first attempts to solve this problem was by producing a volumetric display with 20 focal planes in 2003 (Sullivan, 2003). Akeley et al., produced a similar display with 3 focal planes, and conducted a study on user perceptions of objects with different depth cues (Akeley, 2004). Another display by Schowengerdt and Seibel was designed to allow for dynamic shifts in both accommodation and vergence (Schowengerdt, 2004). In 2008, a similar prototype with 4 different focal depths was developed by Kim et al., and accommodation results were measured using an artificial eye composed of a pinhole and multiple lenses (Kim, 2008).

A more recent HMD type prototype display was developed using liquid lenses, providing addressable focal planes from as close as 8 diopters to infinity as well as variable focal depth (Liu, 2010). One of the most recent attempts at creating a multi-focal plane display was by Maimone et al., but displaying content correctly in different planes is computationally intensive (Maimone, 2013). Several other studies exist that evaluate perception and outline new display designs (Cho, 2012, Hu, 2014, Kim, 2011, Liu, 2010). Though eye tracking is not utilized in most of these studies, the results of experiments involving depth judgment suggest that multiple focal planes can potentially be utilized for interaction in future research (Liu, 2010). With improved methods for reproduction and perception of images, the opportunity has arisen for focus to be used as a means to automatically manage content.

One method for using gaze and depth for interaction in a static 3D display was developed by Kwon et al. in 2006. The system used a parallax barrier type stereo display positioned 84 centimeters away from the user, and was able to estimate depth from a user's gaze on 16 different regions of the screen (Kwon, 2006). Another application by Lee et al. used gaze and blink interaction for annotation tasks in a marker based AR workspace, though the display only utilized a single focal plane and focal depth was not considered (Lee et al, 2010). 3D gaze has recently been proposed as a method for monitoring human attention (Ki et al., 2007), which opened up new opportunities for gaze to be used in other attentive interfaces.

Although focus has previously been proposed for interaction with text such as the system proposed by Toyama et al., research up until now lacks interaction methods for multi-focal plane HMDs (Toyama, 2013). Despite the appearance of several multi-focal or vari-focal HMDs, studies with those displays are limited to depth perception and have yet to take advantage of focal depth via eye tracking. Here, I describe the first study that 1) measures accuracy and variance of focus in a monoscopic, multi-focal HMD and that 2) tests the feasibility and accuracy of automated methods for interaction in monoscopic displays.

4.3 System Design and Framework

Taking the previously mentioned challenges into account, I set out to build an interactive prototype, test its potential for focus based interaction, and develop a framework to facilitate both automated and manual interaction methods. I first construct a 3D gaze tracking system combined with a multi-focal plane HMD that does not require the use of external tracking or projection hardware. Next, a framework is developed to facilitate more natural interaction with elements at varying focal distances and propose various methods for automating display of virtual content. I then conduct a series of tests on focal accuracy and depth cues in the prototyped display to determine the viability of the proposed methods, the results of which are discussed in the experiments section.

4.3.1 Multi-focal Plane HMD Prototype

Since most 3D display prototypes are static and cannot be used for mobile AR, I selected an HMD form factor for this prototype. It consists of an array of three 800 by 600 pixel AirScouter displays, each with its own digital input and depth control. For each plane, the focal depth can be set from 30 centimeters (cm) to 10 meters (m). The three displays were lined up so that three images could be viewed simultaneously during the experiment. The number of planes and their corresponding distances were selected via a pilot experiment and since other research has also been conducted on information presentation in the near, mid, or far visual fields (Uratani et al., 2005). Also, the larger the number of focal planes, the harder it is for users to distinguish between them. The focal distances were set at 30cm for near-field, approximately 1m for mid-field, and at 10m for far field using the manual depth controls on each display. These distances are similar to several other static display setups (Kim et al., 2011, Liu et al., 2010).

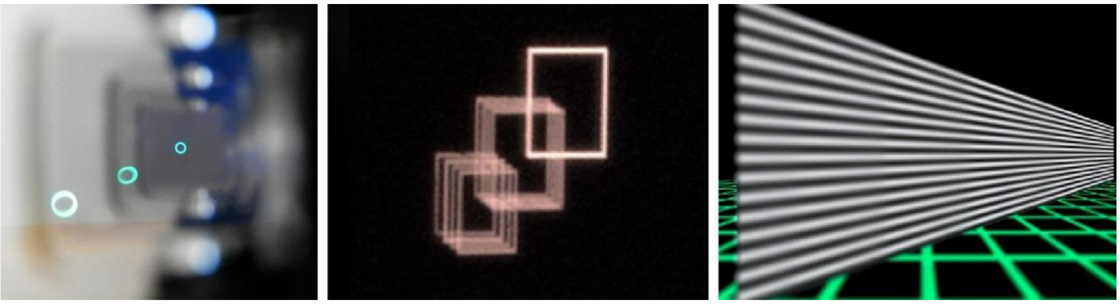


Figure 21 View through the multi-focal plane HMD prototype showing circular icons (left), through Kim et al.'s slanted light source display showing rectangular icons (center, Kim, 2011), and a simulation of accommodation effects by Liu et al. based on their guidelines for designing depth-fused displays (right, Liu, 2010).

For reference, Figure 21 shows views through the HMD, the display designed by Kim et al., and a simulation of accommodation effects for the multi-focal plane display design proposed by Liu et al. Secondly, I needed an apparatus for eye and vergence tracking that could be used simultaneously with a head mounted display placed near the user's eye. In order for focal depth to be measured appropriately, a user's eye convergence must be consistent and eye tracking hardware must provide enough accuracy to correctly select a target icon in the proper focal plane. In addition, I needed a way to make sure that the distance between the tracker and HMD would remain the same for every user. To ensure these conditions, a pair of SMI Eye Tracking Glasses was used as shown in Figure 22 A, and created a 3D printed fastener that fixed the distance between the prototype HMD and the eye tracker as shown in Figure 22 B and C. Though the system still needed to be adjusted slightly for height and width of each participant's eyes, the distance between tracker and HMD remained constant.

4.3.2 Interaction Methodology

Although this prototype can be used for a number of purposes, I first sought to design two natural interactions that have the potential to improve the safety of future wearable displays. The first method was created to intelligently dim or close virtual content when a user changes his or her gaze to an environmental object. The second method is designed to allow a user to manually re-open or interact with previously closed virtual content in a manner that is natural and that incurs minimal distraction. Note that these methods are different with respect to their interaction requirements due the nature of the automation and manual eye movements. Automated dimming of content takes advantage of the natural tendencies of the eye, whereas a manual selection requires a conscious action from the user as well as controlled focus.

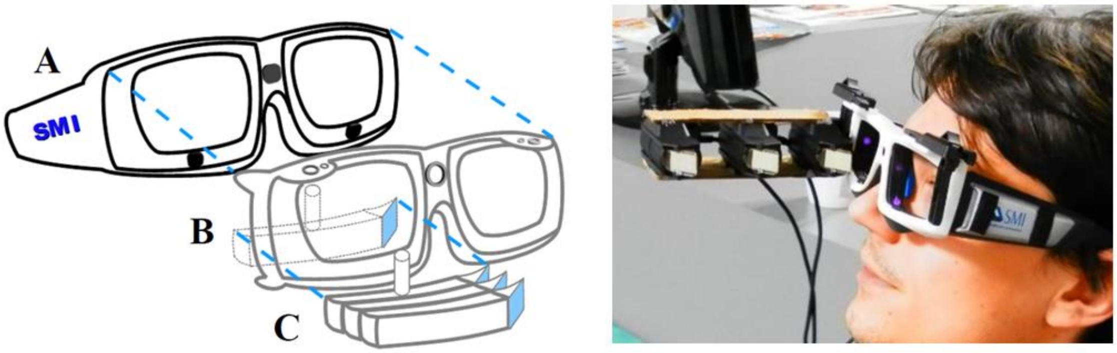


Figure 22 A diagram of the hybrid eye-tracker / HMD prototype showing A) SMI Eye Tracking Glasses, B) 3D printed attachment to affix tracker to HMD and C) multi-focal plane HMD (left). A profile image of a user wearing the prototype (right).

As a user looks away from virtual content and at the environment, physical objects of interest, or oncoming traffic, virtual content should automatically be removed from the field of view to reduce distraction. To accomplish this, focal depth can be used since the eyes start refocusing soon after switching gaze to a new location. As soon as a user's focus leaves the focal plane containing virtual content, I remove text from the screen. For example, if a user changes his or her focus from virtual text at 1m away to a car that is approaching at 10m away, content would be dimmed as soon as the user's gaze leaves the focal range of the virtual text. This will occur as soon as the user's calculated gaze depth exceeds the limit of the current focal plane discretization. In the case of a user walking through a city gazing at building annotations at 30m, text would be dimmed if he or she were to change focus to pedestrians or obstacles on a sidewalk at 1m away.

Once the automated dim or close occurs, the user needs some way to re-engage the dimmed object. Completely removing content from the screen would likely require the user to resort to a physical button press. Instead, I propose that a non-invasive virtual marker, such as one of those used in the experiment described below, remain on the screen. To represent closed or dimmed content, this virtual icon can be left in a corner of the screen and still allow the user an occlusion-free view of other real world objects. By gazing at the aforementioned icon, text or annotations can be re-displayed by executing a dwell or blink action. The user would gaze at the icon at the appropriate focal depth, and the previously dimmed content would be reactivated. The requirements for this type of interaction are somewhat different than the automatic dim or close method since timing and accuracy are more relevant to a successful selection.

In contrast with automatic dim/close, where a user changes focus from a very specific area to anywhere in a different focal plane, manual select requires the user to focus on a more specific area for a certain length of time. In this sense, manual select experiences more of the typical timing and delay problems associated with virtual interaction.

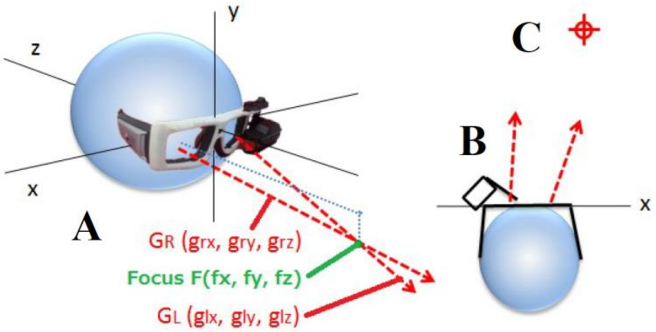


Figure 23 A) Visual representation of depth calculation using intersecting vectors, B) an example of far-field gaze vectors that do not intersect, and C) distant focal point (Toyama, 2014).

Still, whereas a gaze only or line of site method may select virtual content despite a user looking at a physical environmental object in the same direction, incorporating focal distance can be used to eliminate false positives. Moving gaze back to both the direction and focal plane of the virtual icon can ensure that the user desires to re-engage the text and is not merely looking at other nearby content at approximately the same focal depth or in the same line of sight.

4.3.3 Calculating Depth from Gaze

Up to now, a number of different models for calculating gaze depth have been proposed, though few have been designed for multi-focal plane HMD systems (Wabirama et al., 2012). In several informal experiments, gaze accuracy was tested using estimation of vector intersection as well as estimation from raw gaze data based on distance between pupils. This resulted in the selection of two models for calculating depth out of three tested methods, which are described below.

From the eye tracker, a 3D vector of the direction of each eye is first extracted, represented by $\mathbf{G}_R = (g_{rx}, g_{ry}, g_{rz})$ and $\mathbf{G}_L = (g_{lx}, g_{ly}, g_{lz})$ in A (Toyama, 2014). Using this data, the first type of tested estimation was based on linear gaze depth, which is the intersection of the two gaze vectors in 3D space. Unfortunately, these vectors rarely exactly intersect at a single point due to imperfections in the muscles and nerves that govern human eye movement. During several informal experiments testing gaze estimation models, it was sometimes impossible to calculate the point on which the eyes converged in the far-field, since the angle between the vectors was obtuse, as shown in B and C of Figure 23. Furthermore, this method often produces a large error even with a small difference in angle values when the focal point is in the far visual field.

Therefore, instead of calculating depth using the intersection, a regression model of focal depths based on the x-value (g_{lx}, g_{rx}) of two gaze vectors is trained. In short, by comparing the current gaze vector to a number of previously saved gaze vectors for each focal plane, we can achieve a more accurate depth estimate. Since the y axis in Figure 23 A is perpendicular to the line of sight, we can safely assume that the g_{ry} and g_{ly} are always the same. Suppose we have training data represented by

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n),$$

where $\mathbf{x}_i$ is the *i*th element from the training data. Vector $\mathbf{x}_i$ is represented by x-values of both gaze vectors, i.e.,

$$\mathbf{x}_{\boldsymbol{i}} = (g_{lxi}, g_{rxi}),$$

and y_i is the depth value for *i*th training data. A Support Vector Regression (SVR) is trained for the model, which computes a gaze depth value according to given gaze vectors. This SVR functions similarly to vector intersection, but accounts for gaze vectors that may be parallel, obtuse, or non-intersecting.

After training the SVR, we obtain the depth estimation function for a vector **x** given by

$$f(\mathbf{x}) = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) k(\mathbf{x}_i, \mathbf{x}) + b,$$

where α_i, α_i^* are the Lagrange multipliers for the *i*th sample, and $k(\mathbf{x}_i, \mathbf{x})$ is the kernel function.

Though an SVR can be used to calculate a depth value for any given gaze data, the estimation may be inaccurate. If the task is only to discretize the plane at which the user is currently looking at, I can consider this a multiclass classification problem. When I classify user gaze depth into one of multiple focal planes in the prototype, support vector machines (SVMs) are applied. The advantage of this method is that it becomes possible to distinguish focal planes even if a precise focal distance cannot be calculated. In other words, even if I have noisy gaze data or if calculated depth varies between users, the plane in the prototype on which a user is focused can still be determined. In other cases, an SVR is applied when a linear gaze depth value is necessary.

4.4 Experiments

Experiment goals included 1) testing the resolution at which focal depth could be tracked and 2) testing the ability of users to focus on interactive elements to determine if the interaction framework was feasible. These were evaluated through two separate experiments, including a pilot study where ideal focal depths were determined and a more in depth study that utilized the findings from the pilot experiment.

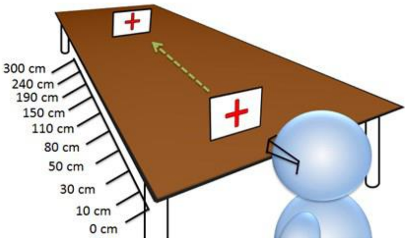


Figure 24 Pilot experiment setup showing gaze targets and depths at which measurements were taken.

4.4.1 Pilot Study

Here I briefly describe the results of a pilot experiment with 4 users testing focus on physical objects at different distances. The resulting data allowed for calculation of appropriate distances between focal planes in the prototype.

4.4.1.1 **Setup**

To get some idea of how accurately gaze depth could be calculated, participants were tasked with focusing on a plus symbol on a small sheet of paper and rotating their heads horizontally from left to right in a 180 degree arc. Participants sat with their eyes at the level of the plus as shown in Figure 24. The eye-tracking apparatus was affixed to the participant's head, and the task began. The participant was asked to focus on the plus at 10cm and rotate, and approximately 10 seconds of gaze data (at least 200 samples per user) was recorded. This process was repeated at 30cm, 50cm, 80cm, 110cm, 150cm, 190cm, 240cm and 300cm, and then repeated for each participant. For the SVR, recorded gaze data was separated for testing and training. A 10-fold cross validation was processed for evaluation. For the kernel function, a radial basis function (RBF) from LibSVM (Chang et al., 2011) was utilized.

4.4.1.2 **Results**

Resulting rotation data for one user is plotted on the left of Figure 25, which shows gaze samples at each distance and trend lines representing feasible, relative depth estimations up to 190cm as a function of head movement. Despite rotation, accuracy of depth estimates remained relatively constant in the near and mid viewing fields. Humans typically turn their heads if gaze angle exceeds 30 degrees, making data outside this range less relevant. More importantly, accurate separation of focal planes becomes difficult after 110cm. A plot showing depth estimation for each participant at each focal distance is shown on the right of Figure 25. Each cluster of four points shows both the estimated distance as well as variance at that depth for a single user.

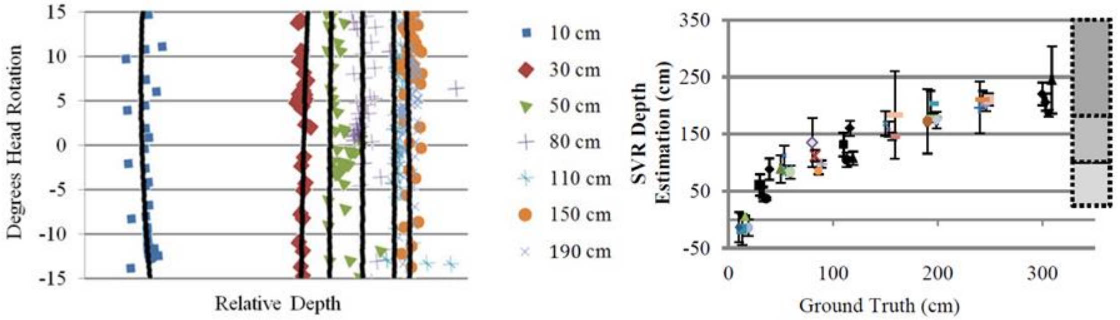


Figure 25 Gaze depth estimates for approximately 30 degrees of head rotation, where y axis is rotation and x axis is relative linear depth. Trend lines are shown for reference (left). Gaze depth estimates with SVR for all 4 users and respective standard deviation. Resulting focal plane discretizations are delimited by the gray dotted boxes (right).

From this data, I needed to select the focal distances with the smallest overlap, which would consequently maximize the chance a focal plane in the prototype would be correctly selected, even with noisy data. Minimal overlap occurred at the three depth ranges shown by the gray dotted boxes on the right hand side of Figure 25. Based on these experiments, a near, mid and far plane, with distances at approximately 30cm, 1m, and anything further than 2m, respectively, were utilized. As expected, gaze data differs between subjects, making precise depth estimation difficult for practical use, however, the method of discretizing focal planes provides significant robustness to variable data. As evident in the next experiments testing depth based selection, these distances were appropriate choices.

4.4.2 Pilot Study on Refocus Timing

The first pilot experiment confirmed that trackable focal distances were limited to approximately three planes, so I then sought to answer the following two questions: 1) How do changes in accommodation between virtual and physical objects affect task performance, and 2) does accommodation affect performance differently at different depths?

4.4.2.1 **Setup**

This experiment was conducted with a single plane AiRScouter HMD, but with the focal depth of virtual text set at three different distances between trials. Eight participants (6 male and 2 female, average age 25.0 years), none of whom wore glasses, were tasked with typing numbers that appeared in the display, and as soon as a number was correctly pressed, a new number would appear in a random location on the screen, as shown in Figure 26. Participants were instructed to use a single finger for the typing task on the upper row of numbers of a full sized keyboard so that they could not use memorized key positions, of the right side keypad for example. This ensured that users had to intermittently switch focus between the virtual number on the display and the keyboard. The first half of the experiment involved a set of 3 tasks, each four minutes long.

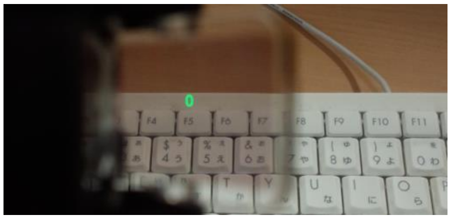


Figure 26 Image through the AiRScouter showing the virtual text and keyboard.

Though the number of trials varied based on the speed at which users pressed the keys, each user completed between 200 and 300 key presses during each of the tasks. The focal distances were set at 30cm, 60cm (to match the focal plane of the keyboard), and 1m, and were randomly ordered and balanced between participants. Thirty and fifty point (pt) font sizes were presented, representing viewing angles of 0.82 and 1.22 degrees, respectively.

The second half of the experiment was a similar set of three tasks with the same set of focal planes, but font size was selected randomly from values between 20 and 80 pt to see whether performance for different font sizes was affected by focal distance.

4.4.2.2 **Results**

For the first set of three tasks, a significant difference in performance and error rates was found between the 60cm and 1m focal planes, but there was almost no difference between the 30cm and 60cm distances. This is most likely due to the fact that distance estimation of objects at greater focal depths is more difficult in general. Median reaction times with standard deviations and error rates for 30cm, 60cm, and 1m, at 30pt and 50pt fonts are shown in Table 5. A one factor analysis of variance (ANOVA) showed a significant effect of focal plane for reaction time and error rate for both 30pt ($F_{reaction(2,21)}$=7.08, P<.01), ($F_{error(2,21)}$=5.45, P<.02) and 50pt ($F_{reaction(2,21)}$=3.59, P<.05), ($F_{error(2,21)}$=3.76, P<.05) at 50pt. The second experiment also revealed an increase in delay for smaller font sizes for the 1m focal plane, as shown in Figure 27.

Table 5 Results of the pilot experiment showing median reaction times in milliseconds (with standard deviation) and error rates for different fonts at different focal depths.

	30 pt font		**50 pt font**	
Focal Depth	Reaction Time (Stdev)	Error Rate	Reaction Time (Stdev)	Error Rate
30cm	1026.88ms (236.2)	3.72%	953.44ms (144.7)	1.63%
60cm	1098.05ms (103.194)	2.64%	996.47ms (130.7)	1.37%
1m	1414.49ms (172.3)	22.21%	1155.21ms (114.4)	9.25%

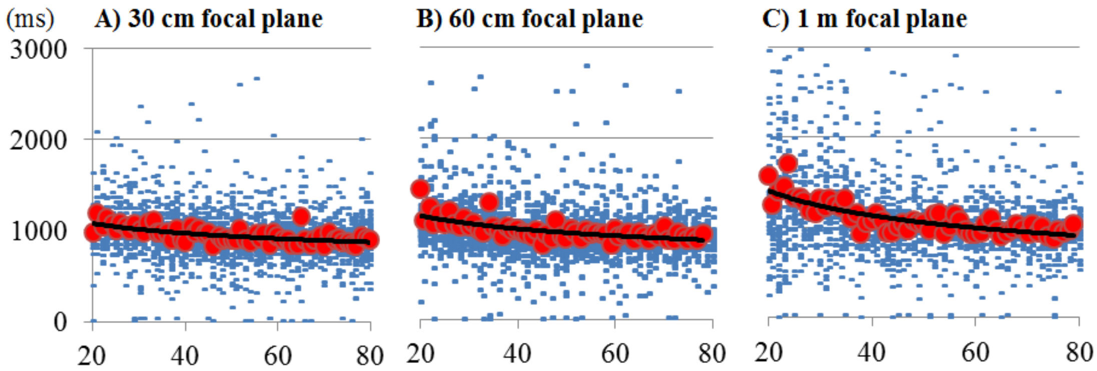


Figure 27 Reaction times versus font size for individual key presses at each focal plane (A, B, and C). Vertical axis is in milliseconds and horizontal axis represents font size. Blue dots represent individual press reaction times, and red dots represent average reaction time for a particular font size. Trend lines are added for reference.

Though A) and B) had no significant difference, C) shows significantly slower reaction times for smaller fonts. This data suggests that larger differences in the focal plane of the physical and virtual reading task are compounded for more intricate tasks in the virtual interface. Consequently, displays that utilize smaller font sizes may result in slower reaction times to events in the physical world. This serves as good motivation for the automated dimming method, since quick removal of distracting text will immediately provide a clear view of the environment. Further testing on the automatic dimming showed that text can be re-engaged with eye tracking even when dimmed (Toyama, 2015).

4.4.3 Study on Icon Selection and Depth Cues

In order to determine whether the automated dim/close and manual interaction methods were feasible, I conducted a second, larger experiment.

4.4.3.1 **Setup**

Users were tasked with viewing a number of different icons through the HMD at different focal depths. I also included various colors and depth cues such as blur and texture to see if there was any effect on variance of physical eye convergence. Since prior research mostly focuses on perception, I wanted to go a step further and learn about the physical behavior of the eye, especially in a monoscopic display.

A total of 14 users, 9 male and 5 female, participated in the experiment. Using the hybrid eye-tracker and HMD prototype discussed previously, 9 sets of 3 different icons were presented to each user. A view through the HMD of one set of icons without any simulated depth cues is shown on the left of Figure 28. Though a number of different icon shapes could have been tested, I chose a circle since Landolt rings and circular icons are often used in visual aptitude tests (Nguyen et al., 2012). Secondly, depth cues such as texture can be displayed symmetrically on all axes, eliminating additional variables. Simulated monocular depth cues were held as variables, and included relative size, texture gradient, defocus blur, and a combination of all three. Figure 28 shows both a simulated 3D view of a single set of icons including all depth cues and two images taken through the display itself, showing the increase of cue strength as plane distance increases. Note that the blue lines in the simulated figure were not visible in the experiment. Cues inherent to the display which were held constant throughout the experiment included accommodation due to focal depth and elevation, since more distant icons were presented at higher vertical locations. Three different colors were also presented, including bright green, fuchsia, and white to test whether certain colors are better focal targets at different depths.

Each participant was first asked to put on the prototype and adjust the HMD until he or she could see all three icons clearly. Next, the user was asked to gaze at one of the three icons, and we recorded 10 seconds of gaze data for that icon. The process was repeated for each icon in the set, the next set of icons was displayed, and the task was repeated for all 9 sets of icons. The order of the 9 sets of icons was randomized between participants to prevent any learning effects. All trials were conducted in a room with constant lighting conditions, and all participants were instructed to face a diffuse, uniform wall throughout the experiment. Individuals that required glasses for significant vision correction were excluded from the experiment since the eye tracking apparatus requires an unobstructed view of both pupils.

4.4.3.2 **Results**

The most important result from this set of experiments was that we achieved a high degree of accuracy for focal plane identification. Out of all samples taken for all users, 98.63% of points were classified into the correct focal plane. Even in the worst case scenario, the sampling data of which is shown in Figure 29 B, we achieved 85.6% accuracy. Since this data is per sample, this means that we can correctly classify a focal plane with near 100% accuracy using a running average of less than 10 samples.

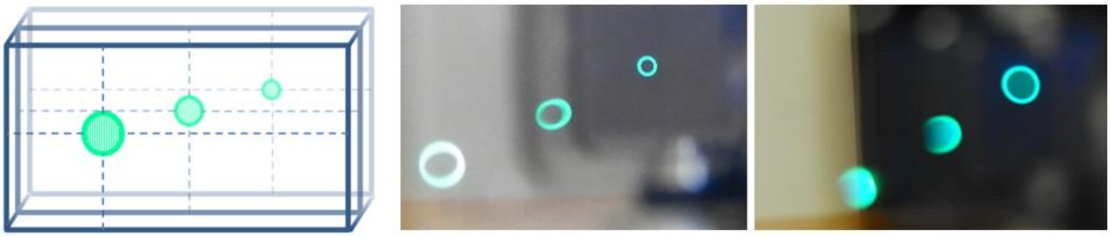


Figure 28 Simulated 3D view of a set of icons including relative size, texture gradient, and defocus blur (left). Pictures taken through the prototype showing two trials, each with three icons at different focal planes and with different depth cues (middle, right). Size is varied on the middle image and texture gradient is varied on the right. The camera is focused on the farthest icon in both.

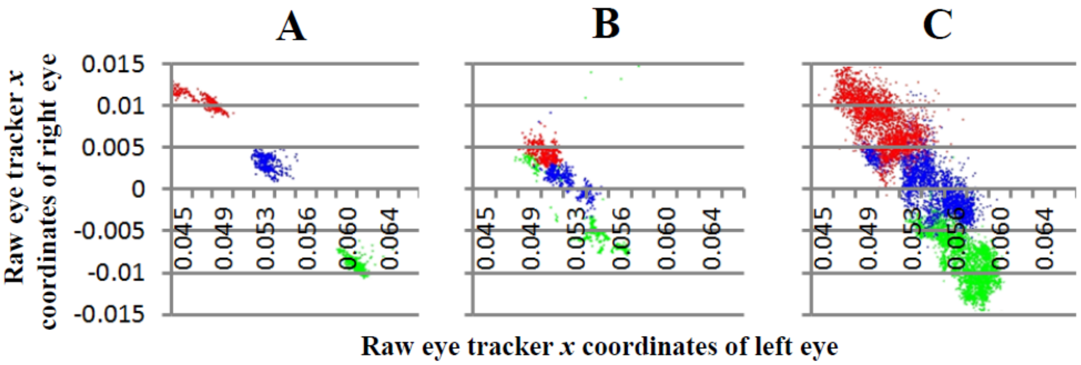


Figure 29 Gaze data for A) a trial with high accuracy (98%+), B) the trial with the worst accuracy (85%) and C) all trials for a user with average accuracy. Focal plane classifications are represented by green (30cm), blue (1m), and red (2+m), and are classified using the raw x coordinates produced by the eye tracker for each eye.

This means that it took less than 500ms worth of samples to identify a correct plane for any user, and less than 100ms for a vast majority of cases. For real time everyday use, this low latency is essential for quickly removing distracting information. Also, based on the sample data for all trials shown in Figure 29 C, it would become increasingly difficult to accurately determine which sample belongs to which plane as the number of planes increases. Hence, 3 focal planes is likely an excellent choice to ensure robust identification.

4.4.4. Commercial Single-plane HMD Usability Study

Now that I had tested interaction in the multi-focal plane case, it made sense to test the method for commercially available devices, since displays that are now available for industrial and consumer use. Simply speaking, I wanted to know whether the same plane classification accuracy could be achieved in different commercial devices with the different focal plane depths if a user switched from a virtual reading task to the physical world, or vice versa.

Table 6 ANOVA of gaze accuracy with both colors and sets of cues held as constants and variables.

Constants	Variables	$F_{(1,14)}$	P-value
Green	no cues, all cues	0.05	0.83 > .05
White	no cues, all cues	0.15	0.70 > .05
Fuchsia	no cues, all cues	0.06	0.81 > .05
Green	blur, gradient, size	0.12	0.95 > .05
No cues	green, white, fuchsia	0.98	0.38 > .05
All cues	green, white, fuchsia	0.12	0.95 > .05

4.4.4.1 **Setup**

To test this, 12 individuals were asked to participate in an experiment in which they read virtual text on both the AirScouter and Google Glass, and switched to various physical targets. In the experiment, participants were provided with a total of 4 gaze targets, including 2 physical printouts positioned at 60cm and 2.5m from the participant’s eyes, and the virtual planes of the HMDs (tested one at a time). The AirScouter was set to a 30cm focal plane, and Google Glass was viewed at its fixed distance, which has an advertised perceived distance of approximately 2.5 meters, though the actual focal distance is unpublished, but estimated at between 1 and 1.5 meters. The displays were affixed to two SMI eye trackers, and their orientations were fixed so that users could read text and eye tracking could be conducted at the same time.

Each participant conducted 8 trials for each device, where one trial consisted of a participant either reading a virtual text and then switching to one of the physical printouts, or starting with a physical printout and then switching to the virtual text. Each physical printout had 5 crosshairs located in the corners and center of the paper. Both printouts were scaled so that all 5 crosshairs would fit into the virtual field of view of either of the HMDs, even with a small amount of head movement. Numbers from 1-5 were located next to each crosshair, and each number was read out loud to direct the participant to switch his or her gaze to the next crosshair. Participants were instructed to keep the physical object and virtual plane aligned during each trial. The virtual task was to read a short paragraph presented in the display.

Auditory cues to start, switch, and stop were given between each 5 second task, and each trial included at least 10 seconds of total gaze data, including one virtual and one physical gaze task. Half of the participants started with Google Glass, and the other half started with the Air Scouter. The order of each 10 second trial, as well as the two 5 second tasks within each trial, were randomized between participants to eliminate any ordering effects. At the end of the experiment, users took a short survey with questions regarding preference and ability to focus and read text on each of the displays. The first four trials of each participant were taken to establish a ground truth, and the remaining trials were run through the classification algorithm to test accuracy on each individual device.

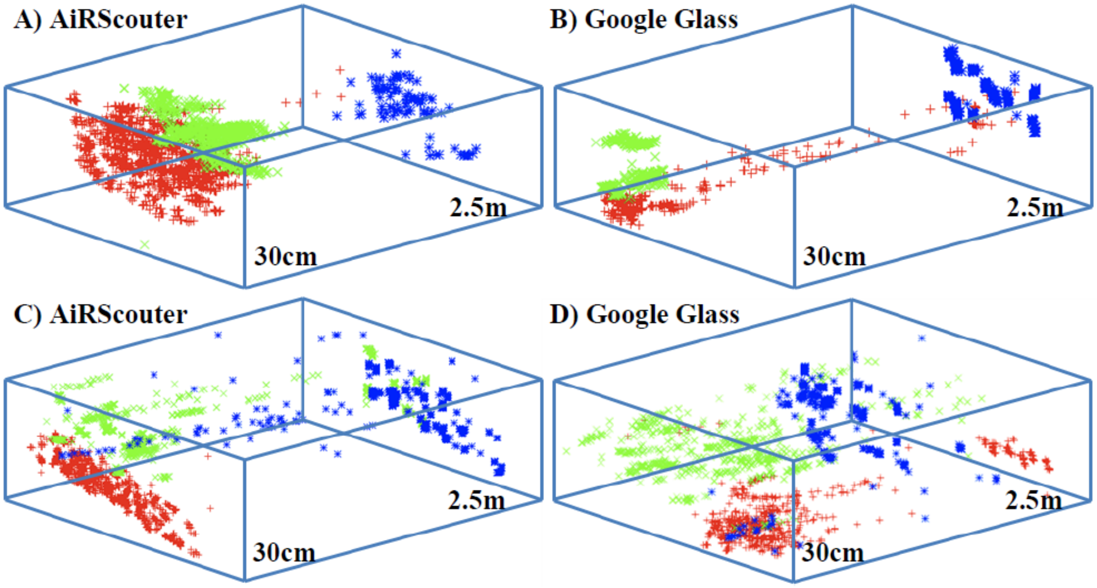


Figure 30 Gaze samples showing results for classification into 3 planes (V – N – F) where participants had relatively consistent gaze (A and B) and less consistent gaze (C and D). The left hand plots show data for the AiRScouter, and right hand plots are for Google Glass. Red, green, and blue samples represent virtual, near, and far planes, respectively.

4.4.4.2 **Results**

For both devices, four types of analysis were carried out to test how well the classifier can separate 1) virtual from real (i.e., the virtual plane from either of the two physical planes), 2) all three planes (i.e. near physical, virtual, and far physical), 3) virtual to near physical, and 4) virtual to far physical. These four cases are discussed below using the abbreviations 1) V - N+F, 2) V - N - F, 3) V - N, and 4) V - F, where V is virtual, N is near, and F is far. The plots in Figure 30 represent visualizations of ground truth data from both good and bad cases for each device. The *x*, *y*, and *z* points plotted in 3D space are calculated using the gaze vectors from the SMI tracker, which reports an accuracy of less than 0.5 degrees angular error for line-of-sight eye tracking. For the AirScouter, gaze consistency can be seen in A and C of Figure 30, represented by the red samples. Consequently, the AiRScouter was correctly separated from both the 60cm and 2.5m printouts for 93.2% of classified samples. In contrast, samples for Google Glass had a much larger deviation for all participants, as shown in B and D of Figure 30. This is likely due to the fact that it is more difficult for humans to judge depth by accommodation alone at farther depths (Van, 2012). According to the post-experiment survey, participants also stated that it was more difficult to focus with Google Glass. Although the average classification accuracy for Google Glass was only 75.7% for the V – N – F case, separating V – F still resulted in 88.1% accuracy.

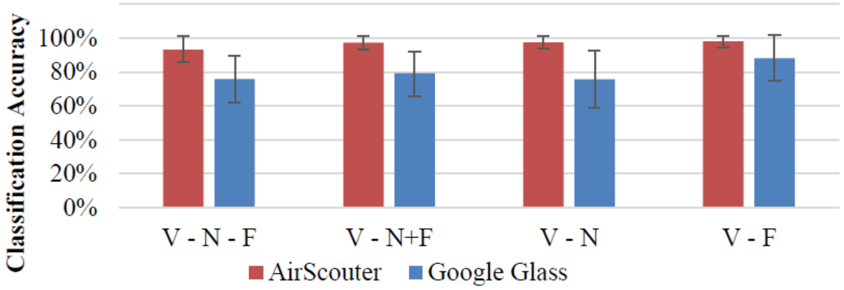


Figure 31 Plot showing four types of analysis for classification accuracy (with standard deviation) by device. V, N, and F represent virtual, near, and far planes, respectively. For example, V - N represents the ability of the algorithm to correctly classify virtual and near planes, when points are only segmented into the virtual and near groups.

This means a majority of users will still be able to use this method for interaction, although people who may have trouble focusing on a distant screen could potentially experience a number of false positives. A plot showing classification accuracy by device and separation case is shown in Figure 31, and a one factor analysis of variance (ANOVA) shows a significant effect of device ($F_{(1,22)}$=13.89, P<.01). From this data, we can conclude that nearer virtual planes can be separated more easily than far, so classification will be device dependent. As such, the classifier can be used well for planes around or under one meter, but may fail for products that have a focal plane past that of Google Glass since users have very little information with which to judge actual depth of virtual content.

4.4.5. Discussion

These findings show that this interface is a good next step in the road towards attentive interfaces, especially for those that are used in mobile situations requiring divided attention. One such example is that of a medical assistive interface for doctors or patients (Ames, 2004, Sonntag, 2013). In the case of the patient, we can utilize focus to drive assistive visualizations such as environmental navigation labeling for a patient who may be lost. Focus also has the potential to be used for cognitive impairment modelling or detection, for impairments that are related to eye movement and control. For this purpose, the SVMs we proposed in the interface design section may prove to be useful. Most importantly, I found that tracking of focus on icons, reading tasks, and real world gaze tasks can be separated with high accuracy for use with intelligent user interfaces. In most cases, gaze depth could be calculated with only a few samples, which means that automated dimming or closing of content can be executed almost instantly. This speed is essential for times when a user may have to react to an oncoming car or hazardous object.

Another interesting finding from the experiments is that eye convergence occurs even when an image is presented only to a single eye in a monocular display with multiple focal planes. Though convergence is much less accurate, both eyes still move to a point of interest in 3D space, even without presentation of a stereoscopic image. It is very likely that this effect is tied to the same mechanism that controls blinking and pupil dilation, considering both eyes will respond when presented with blink stimuli or light to only one side. We can safely assume that classifying focal plane using eye tracking be accomplished in a monocular HWD with multiple focal planes. As the distance of the focal plane increases, classification becomes increasingly difficult. It is still unclear as to whether these results will hold true for three virtual objects of the exact same size and in the same line of sight. However, it is very likely that the human brain can easily separate the focal depths of objects with different properties or colors, such as a physical car and virtual object.

Another unusual finding is that the depth cues in the experiment and variance in physical eye convergence were largely unrelated, despite a strong demonstrable relationship between depth cues and depth perception in other research. This evidence suggests that accommodation, the constant monoscopic depth cue in the experiment, may be the stronger factor when human eyes try to converge at a certain depth. Though only three highly visible colors were selected, multiple colors and shapes also deserve consideration.

4.5 Summary

In this chapter, I first conducted several initial tests to explore the potential of focus based tracking for interaction in a monoscopic display, and found that switching focus from near to far planes can affect performance and that gaze depth can be tracked with limited accuracy. I then designed an attentive interface that can automatically separate virtual and environmental gaze patterns based on knowledge of focal plane depth. After facilitating automated content removal in both single and multiple focal plane displays, the range of interaction with virtual icons and text is tested, and I found that this type of interface has the potential to effectively manage text in real world situations when only monoscopic depth cues are present.

CHAPTER 5

View Manipulation

AR devices have great potential to enhance or improve natural human vision. However, several problems still exist that prevent current displays from providing a wide field of view or easy engagement of augmentative functionality. Secondly, most displays lack an easy, intuitive way to engage augmentative functionality and a flexible way to interchange augmentations. In this chapter, I will introduce the Fisheye Vision and ModulAR displays, which are designed to deal with some of these problems. Experiments testing each device are also described and discussed.

5.1 Introduction

One current problem with many video see-through displays is the lack of a wide field of view, which can make them dangerous to use in real world augmented reality applications since peripheral vision is severely limited. Existing wide field of view displays are also often bulky, lack stereoscopy, or require complex setups. To solve the problem of limited FOV, I introduce a prototype that utilizes fisheye lenses to expand a user's peripheral vision inside a video see-through head mounted display. The system provides an undistorted central field of view, so that natural stereoscopy and depth judgment can occur. The peripheral areas of the display show content through the curvature of each of two fisheye lenses using a modified compression algorithm so that objects outside of the inherent viewing angle of the display become visible. I first test an initial prototype with 180 degree field of view lenses, and then build an improved version with 238 degree lenses. I also describe solutions to several problems associated with aligning undistorted binocular vision and the compressed periphery, and finally compare the prototype to natural human vision in a series of visual acuity experiments. Results show that users can effectively see objects up to 180 degrees, and that overall detection rate is 62.2% for the display versus 89.7% for the naked eye.

5.1.1 Improving Peripheral Vision with Spatial Compression

In recent years, head mounted displays (HMDs) have finally achieved a form factor that allows them to be worn comfortably for long periods of time. Products like Google's Glass, Epson's Moverio, Vizux's Wrap, and Oculus's Rift are becoming commercially available, and increasingly commonplace. However, a number of problems with these devices remain. In particular, the narrow FOV of most see-through displays poses a problem to user safety when conducting simple tasks like walking, navigating, or checking for oncoming traffic. Current solutions to this problem include prototypes designed to provide a wide FOV, but are often bulky or do not provide good binocular vision (Ardouin et al., 2012, Chen et al., 2002, Kiyokawa, 2007, Nagahara et al., 2006). In the case of video see-through displays, problems like limited resolution, pixel persistence, narrow field of view (FOV), and delay can make devices unsafe to use for everyday augmented reality (AR) applications.

As a next step, I propose the use of fisheye lenses to expand a user's peripheral field of view for both general and outdoor augmented reality (AR) applications. I developed a setup that somewhat resembles other stereo AR displays such as those by Kiyokawa et al and Fan et al (Kiyokawa et al., 2007, Fan et al., 2014). However, my prototypes include several major differences, including the use of ultra wide angle fisheye lenses and modified undistortion algorithms for images in the peripheral region of the display. The prototypes, which are modified versions of the Oculus Rift, are shown in A and B of Figure 32. In my design, binocular vision is achieved by undistorting the pixels in the central field of view, as in other models. The big difference is that images presented in the peripheral view are shown to the user as if viewed through a fisheye lens, as can be seen through the left eye camera in D of Figure 33, but with several modifications.

Wide angle lens distortion would introduce a number of problems such as reduced depth perception and skewed direction estimation if it were in the binocular field (Brandt et al., 1973, Kruijff et al., 2010, Watson et al., 1995). Based on the results of these studies, it made sense to avoid major rotations or changes in scale when designing the peripheral compression methodology. In contrast, my prototype provides binocular stereoscopy and a simultaneous compressed view of the peripheral, allowing users to constantly view objects up to 180 degrees. Furthermore, I conduct a number of studies on perception and visual acuity of compressed objects displayed in the periphery, such as the effect of lens compression on reaction time. Additionally, since acuity in human peripheral vision is already low and does not have a binocular component, a fisheye view of the periphery can escape many of these problems. The great benefit of this expanded view is that objects such as cars and pedestrians that are beyond the HMD screen's angular viewing plane become visible to the user. This means that peripheral objects of interest come into view more quickly, and in most cases are noticed at angles similar to those of normal human vision.

Additionally, the prototype is lighter than most helmet-based and catadioptric systems, is inexpensive to construct, and can be used for outdoor AR for extended periods of time. In the following sections, I describe the detailed setup of an initial prototype using 180 degree FOV lenses and an improved design using 238 degree FOV lenses. I then present the results of a series of experiments testing a user's ability to notice peripheral objects of different sizes and at different angles in the redesigned display. Users conduct the same tasks with both the display and naked eye to provide an objective comparison to natural human vision.

5.1.2 Prior Work

Related research primarily falls into two categories. These include 1) expanding a user's virtual FOV through hardware or software and 2) studying displayed objects and perception in expanded or modified peripheral views.

One of many attempts at expanding a user's FOV in a head mounted display was in 1999 by Yamazaki et al. They prototyped a prism based OST display that offered a 51 degree wide

FOV (Yamazaki et al. 1999). Another recent attempt to accomplish a wide FOV using projective displays was carried out by Kiyokawa et al. This display was developed using hyperbolic half-silvered mirrors in combination with a retro-reflective screen, which gives users optical see-through capability (Kiyokawa, 2007). Subsequently, a number of other design guidelines and display prototypes were created that used mirror and lens systems to expand the physical FOV to the periphery (Chen et al., 2002, Shum et al., 2003). In 2006, Nagahara et al. developed a VST display that converts the image from a 360 degree catadioptric camera system into two stereoscopically aligned images (Nagahara et al., 2006). These images, which compensate for distortion, are subsequently projected onto two hemispherical lenses, and provide a near 180 degree field of view. Most previous designs are relatively bulky and often require separate projectors and mirrors for each eye.

A similar display proposed by Ardouin et al. in 2012 also uses a catadioptric camera to compress 360 degree of viewing field into a 45 degree FOV display (Ardouin et al., 2012). Unfortunately, this introduces significant distortion into the user's binocular vision, and only a short quantitative experiment was carried out. To my knowledge, the most recent attempt at providing an expanded field of vision is that of Fan et al. in 2014 (Fan et al., 2014). They present a single 100 degree wide field of view camera image to both eyes (biocular view). Instead of a user being able to view his or her peripheral environment, a number of different indicators are blended into the displayed image to indicate objects of interest. In contrast, my prototype provides binocular stereoscopy and a simultaneous compressed view of the peripheral, allowing users to constantly view objects up to 180 degrees. Furthermore, I conduct a number of studies on perception and visual acuity of compressed objects displayed in the periphery, such as the effect of lens compression on reaction time.

Most past studies on virtual peripheral vision in wearable displays have been limited due to physical restrictions of display technology. However, a number of studies are available that examine various projected objects or modified physical peripheral views in non-virtual environments. Human peripheral vision has been very widely studied, with one of the first relevant studies from Brandt et al., who showed that rotations of the periphery result in a perceived self-rotation (Brandt et al., 1973). This type of perceptual study has been extended into the virtual domain, such as the work by Draper et al., which showed that changes in scale can lead to simulation sickness in virtual displays (Draper et al., 2001). Based on the results of these studies, I sought to avoid major rotations or changes in scale when designing the compression methodology.

More recently, researchers have begun to consider virtual displays for the modification of the periphery. For example, Vargas-martin et al, used an HMD to add peripheral information to the central field of view to help patients with severe tunnel vision (Vargas-martin and Peli, 2002). A more recent study by Loomis et al in 2008 studied perceptions of gaze in human peripheral vision. It was discovered that, to some degree, humans can determine the gaze direction of an onlooker despite the fact that the onlooker's face is in the periphery (Loomis et al., 2008). Even more recently, the predator-prey vision metaphor has been proposed as a method for modifying the periphery by varying the camera angle to simultaneously increase the peripheral FOV while decreasing the binocular FOV (Sherstyuk et al., 2012). My model tries to avoid this modification of camera angle to ensure the user has a more natural and consistent binocular view, but can still reap the benefits of an expanded periphery. Annotation discovery rate has also been studied in wide FOV optical see-through displays by Kishishita et al (Kishishita et al., 2013). This provides further evidence that effective use of both binocular and peripheral view spaces is essential when users need to notice objects beyond the binocular field of vision.

Up to now, a number of catadioptric and view modification systems have been proposed to expand a user's field of view, but these attempts do not always provide good binocular stereoscopy, which is desired for correct projection and augmentation in real-world AR (Bimber et al., 2005).

Other existing studies have yet to compare the apparent benefits of these prototypes to human vision (Ardouin et al., 2012, Nagahara et al., 2006, Veas et al., 2012). In comparison to catadioptric displays, my design has a smaller form factor and requires less hardware. Additionally, problems associated with binocular display techniques have been well studied, but only recently have portable wide FOV displays become commercially available. This allows us to conduct improved studies of the virtual peripheral field, and take advantage of the pixels in the display that are in the periphery.

5.2 Fisheye Vision System Design

To build a usable prototype, I wanted a lightweight, portable display that had at least an 80 degree horizontal field of view. Secondly, stereo cameras had to have an appropriate frame rate and wide enough field of view to match the opening of the fisheye lenses. To provide a decent initial FOV, I selected the Oculus Rift, primarily for its 90 degree horizontal viewing angle. This allows us to utilize 60 degrees of binocular vision, and the remaining 30 degrees of peripheral vision for each eye. It is in these remaining 30 degree sections that I compress approximately 60 degrees of peripheral vision per eye. Depending on the user's exact range of binocular vision, different angular ratios can be used.

The first prototype, shown in A of Figure 32, was designed with two 180 degree FOV wide angle lenses, and was intended to expand a user's vision to 180 degrees, though I later learned that wider angle lenses are necessary. These lenses and web cameras can be purchased for under $50 each. As already mentioned, my setup is similar to some other emerging AR stereo rigs in the sense that I provide binocular vision.

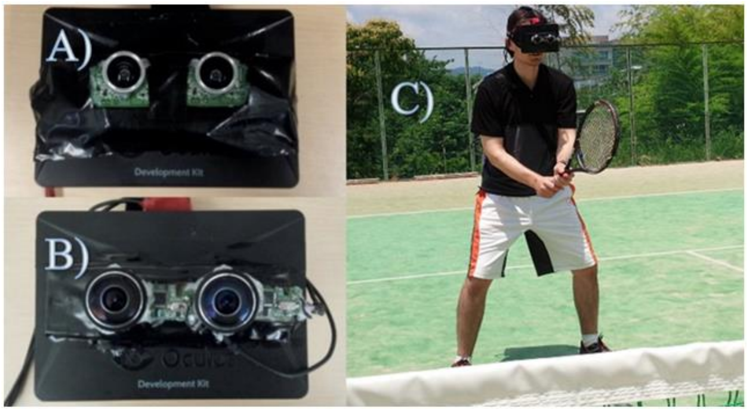


Figure 32 The A) first and B) second prototypes of the Fisheye Vision display with 180 degree and 238 degree FOV lenses, and C) testing of the 238 degree prototype in a tennis rally.

However, where most other setups seek to achieve a perfect one-to-one mapping between the environment and each pixel viewed by the user (Li, 2008, Takagi et al, 2000), I provide this exclusively for a user's binocular vision. Regarding peripheral vision, the prototype varies greatly from other setups. Rather than providing a standard one-to-one mapping, I modify the pixels in a user's peripheral vision to look as they are to some extent viewed through the fisheye lens. This presents a number of benefits, such as the ability to notice objects past the inherent FOV of the display. It also raises a number of interesting questions, such as: To what extent will users notice peripheral objects and how does this compare with the naked eye? Will this affect the time it takes to notice an object? Can users complete everyday tasks with relative ease with this modified view?

Since the display design is atypical for see-through displays, a number of problems arose when trying to correctly display the camera images, including several that are not solvable with normal use of existing computer vision functions such as the stereoalign or undistort functions provided by OpenCV (Bradski et al., 2008). Some setups call for vertical alignment of cameras since they align with the Oculus Rift's pixel distribution for binocular vision. This presents a major problem for my design since I am trying to achieve a higher horizontal FOV, and since some parts of the fisheye lens are not visible when aligned with many standard web cameras, as can be seen at the top and bottom of A in Figure 33. Therefore, I opted for horizontal alignment of the cameras, which allows us to neatly fit a majority of the fisheye lens's horizontal FOV into the camera's input. Vertical FOV is slightly cut off due to the fact that the fisheye lens does not fit perfectly, but this is not a problem since I am primarily concerned with horizontal FOV. Though this orientation results in a minor decrease in resolution due to the mapping between camera and display, I gain control of a much wider horizontal viewing angle.

5.2.1 Undistortion Algorithms

The initial methods used for undistortion and compression can be summarized in four distinct steps, including binocular undistortion, peripheral partial compression, misalignment correction, and peripheral linear compression, the latter of which is described along with the second prototype. The binocular view of 60 degrees is corrected using standard OpenCV functionality (Bradski et al., 2008). A camera image through the fisheye lens is first faced toward a checkerboard to obtain camera parameters, as shown in A of Figure 33.

After obtaining radial and tangential distortion coefficients for each lens, the imageundistort function is then applied in real time to both video streams for all pixels located within the binocular FOV. Although the standard undistort function worked well for the 180 degree FOV lenses, getting accurate parameters for the 238 degree lens system was more difficult. OpenCV's standard functionality actually cannot handle a FOV of over 180 degrees, so if a perfect undistortion of all peripheral pixels is required, a different undistortion algorithm would be necessary. Luckily, a majority of the binocular view presented to the user is viewed through the fisheye lens center, where the distortion is less pronounced.

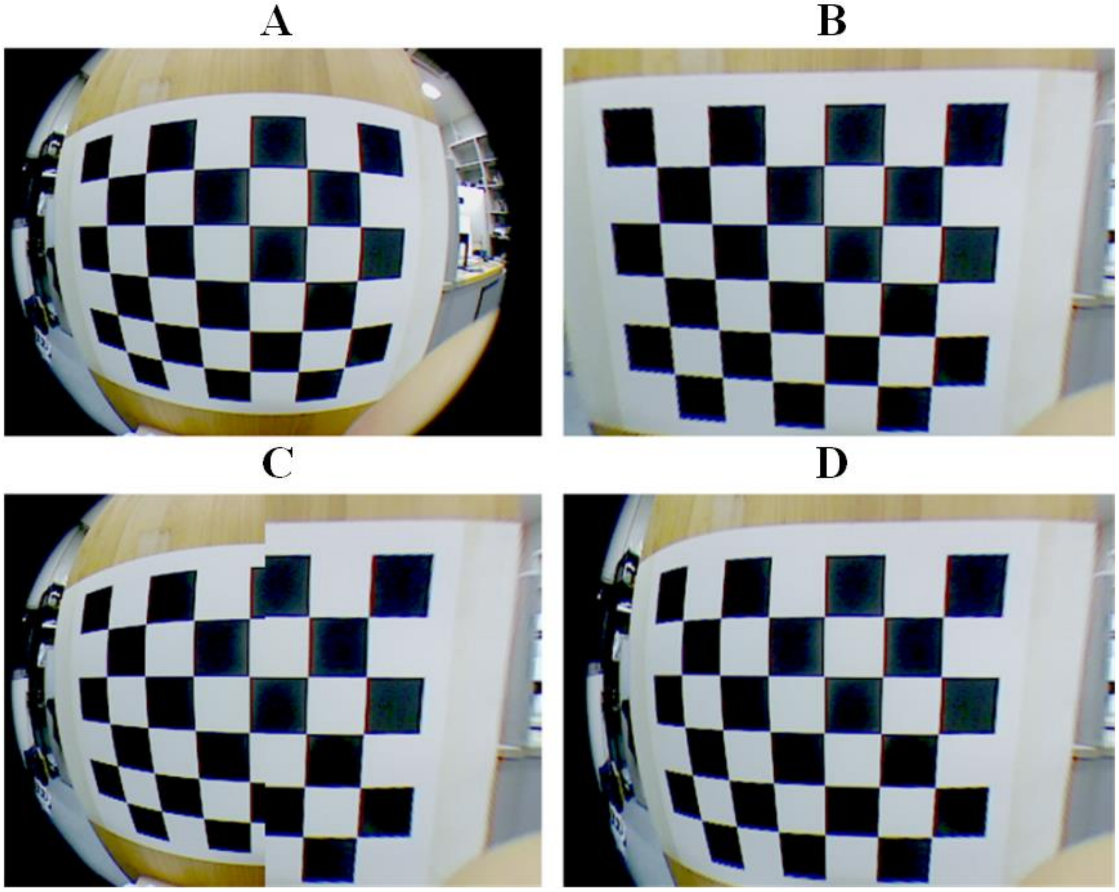


Figure 33 Screenshots from the left-eye camera showing the A) view through the fisheye lens, B) completely undistorted image, C) image using peripheral compression, but with central

misalignment due to variation in compression and undistortion functions (center line), and D) corrected image with modified horizontal peripheral compression.

The more complex part of the design lies in the manipulation of the peripheral FOV. Here, I must effectively compress over 60 degrees of environmental FOV into 30 degrees or less of virtual FOV. First, I left a portion of the virtual image untouched, as if viewed through the fisheye lens. Unfortunately, this results in a very obvious line where the compressed and non-compressed images meet, which is visible in C of Figure 33. So, I was left with an interesting problem: How can I present a compressed image in the periphery and smoothly connect it to the undistorted binocular image? After considering a number of image-stitching and mosaic algorithms to merge the misaligned portion, I found a much more efficient solution, which also provides a more natural view for the user. Instead of running a time consuming alignment algorithm, the undistortion is run using only the *y* values of the coordinate map. This results in both a relatively clean alignment and a less distorted perspective in the vertical domain, as shown in D of Figure 33. In order to accomplish this, I modify the input map to OpenCV's undistort function as follows (Bradski et al., 2008). First, I start with the standard formula used to undistort an image, where (x_p, y_p) represent undistorted points, and (x_d, y_d) are the points viewed by the camera through the fisheye lens. Here, k_1, k_2, and k_3 are the radial and p_1 and p_2 are the tangential distortion coefficients obtained from the checkerboard calibration. The result of y_p is obtained by compensating for radial and tangential distortion using the standard remap function, as in the following

$$[y_p] = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)[y_d] + \left[p_1\left(r^2 + 2y_d^2\right) + 2p_2 x_d y_d\right].$$

The result of x_p is obtained by only compensating for tangential distortion, leaving *x* values in their compressed state, with

$$[x_p] = [x_d] + \left[2p_1 x_d y_d + p_2\left(r^2 + 2x_d^2\right)\right].$$

In code, this undistortion is normally carried out by the *Remap* function,

Remap(*dst, image, mapx_mod, mapy*, *CV_INTER_LINEAR* | *CV_WARP_FILL_OUTLIERS*, *cvScalarAll*(0));,

where *dst* is the destination image, *image* is the original image, *mapx* is the map containing new undistorted point locations in the *x* domain, and *mapy* is the matrix containing new undistorted points in the *y* domain. The modification is accomplished by substituting the default *mapx* parameter with *mapx_mod*, which contains new distortion values. This *mapx_mod* matrix is produced using the undistort function with the radial distortion coefficients set to zero. As a result, the virtual peripheral view still compresses objects horizontally, the vertical ratio of environmental objects to virtual objects becomes closer to one, and the peripheral and binocular images align, as shown in D of Figure 33. Additionally, the complete virtual image more accurately represents the limits of the human field of view, which are more rectangular than circular as can be seen in the differences between the left hand borders of C and D. Note that the border is still somewhat rounded, due to the fact that the *mapy* values are reassigned based on the modified *mapx_mod*. This actually works in our favor, since more vertical content becomes visible towards the edge of the virtual FOV. At this point, I had come up with an effective method for displaying objects outside the native FOV of the display.

However, upon testing the display outside and with a number of different users, I quickly learned that 180 degree (advertised) lenses do not always provide a true 180 degree field of view, partially because the web camera FOV does not perfectly fit the fisheye lens inlets. Additionally, objects towards the outer edge of the fisheye lens appeared extraordinarily small and were barely visible. I then ordered a pair of 238 degree super-wide angle fisheye lenses, which provided a good solution to this problem, and resulted in an improved second prototype, as shown in B of Figure 32.

5.2.2 Improved 238 degree Lens Design

In comparison with the first design, I made three primary modifications to both hardware and software in the second prototype. The first main difference is the use of 238 degree instead of 180 degree FOV lenses. This was a good choice since objects placed around 180 degrees no longer appeared infinitesimally small, and were relatively noticeable on the raw camera image. The second change was with the cameras themselves, which were originally Logitech C500s, chosen since to fit the inlets of the 180 degree FOV lenses. These were upgraded to Logitech C310s, which are more suited to the 238 degree FOV lenses, and provide easier manipulation of exposure and brightness via software. The last change was to the algorithm managing peripheral compression.

Because of the differences between users' interpupillary distances and spacing between the Oculus lenses and the eyes, some people could not see the entire peripheral camera image during informal testing. To compensate for this deficit and in order to conduct more consistent experiments, I applied a small linear compression (equivalent to a perspective change) in addition to radial compression to ensure that data would fill the virtual FOV for all participants. I first compute a perspective matrix *M* using the getPerspectiveTransform (Bradski et al., 2008), which solves for *M* in

$$\begin{bmatrix} t_i x_i' \\ t_i y_i' \\ t_i \end{bmatrix} = M \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix},$$

where x_1 and x_2 are the *x* coordinates representing the vertical division between peripheral and binocular in the display x_0 and x_3 represent the outermost pixel showing content through the display lens. The last computation is done using the previously described (x_p, y_p) as parameters in OpenCV's warpPerspective function, as in

$$(x_f, y_f) = \left(\frac{M_{11}x_p + M_{12}y_p + M_{13}}{M_{31}x_p + M_{32}y_p + M_{33}}, \frac{M_{21}x_p + M_{22}y_p + M_{23}}{M_{31}x_p + M_{32}y_p + M_{33}}\right)$$

where pixels in the final image are represented by (x_f, y_f). All of this processing occurs in about 11ms on a laptop with a Core-i7 3520m processor running at 2.9 Ghz, allowing for display at over 30 frames per second (fps). Although current smartphones may not be able to run the undistortion algorithms at over 30 fps, small form factor laptops or tablets likely have enough power to run Fisheye Vision for mobile AR applications.

5.2.3 Initial Experiments

In the experiments, I sought to evaluate the ability of users to notice objects in the improved prototype and compare this with human vision in terms of both acuity and reaction time. To test this, a number of icons of different sizes and at different angles in each participant's periphery were displayed, and whether or not they were noticed was recorded, as well as the time it took to notice them. The results of this experiment have important implications for safety, since someone using an AR application outdoors that fails to notice an oncoming vehicle may be severely injured or killed. A total of 10 individuals, 3 female, 7 male, all of who willingly volunteered, were tested.

5.2.3.1 **Setup**

The experiment task was to press a button when an icon came into view, and detection rate and reaction time for correctly detected icons for both the 238 degree fisheye display (referred to as the *display* condition) and the naked eye (referred to as the *eye* condition) were recorded. The setup is shown in Figure 34, with a simulated view of all large icons and angles overlaid at the same time for reference. The two projector screens were stationed at 105 centimeters (cm) to the left and right of the user, and a 70cm high table was centered between them. A headrest was fitted and centered on top of the table so that participants' heads would remain at 104cm above the floor.

Directly in front of the participant at a distance of 140 cm was a tablet PC, which displayed random numbers between 0 and 9 every three seconds. Participants read these numbers aloud to ensure that they were concentrating on their central field of view. A single semi-ambient light was positioned behind the user, and luminance was set to approximately 50 lumens (lx) for a blank projector screen, and 55 lx for the wall where the tablet was located, as measured from the headrest. This luminance was selected through informal testing to ensure that a number of icons would likely be missed for both eye and display, allowing for effective observation of differences in error between conditions.

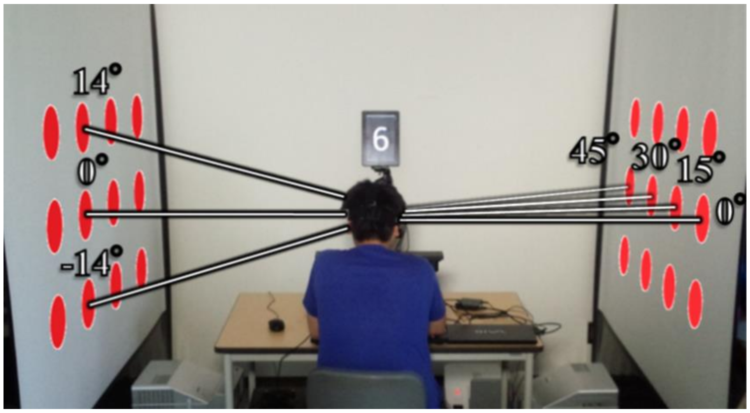


Figure 34 Experiment setup showing projector screens, numbers displayed on the tablet for the concentration task, and a participant. A simulated view of display angles and large icons at every position is overlaid for reference.

Icons were solid red circles and displayed at 0, 15, 30, and 45 degrees in the horizontal, and at -14, 0, and 14 degrees in the vertical, for a total of 12 different positions on each side of the periphery, as designated by the lines and angles overlaid onto Figure 34. Three different circle diameters, 3.5 cm, 7.0 cm, and 10.5 cm were presented, which represent an approximate cone of 1.9, 3.9, and 5.8 degrees of FOV respectively, though perspective was slightly shifted for icons over 0 degrees. For reference, a 1.5 meter wide vehicle at 10 meters away can fit in approximately 8 degrees of FOV, and at 50 meters away, 1.7 degrees of FOV. With these conditions, a large range of object sizes and peripheral locations were covered. Each individual icon was displayed at a random interval between 3 and 8 seconds, and remained on the screen for one second. Conditions were randomized to prevent any ordering effects. The same projector models and settings were used to show images on both screens, and icon positions were calibrated individually to ensure left and right angles were consistent. Each participant completed all tasks in less than one hour.

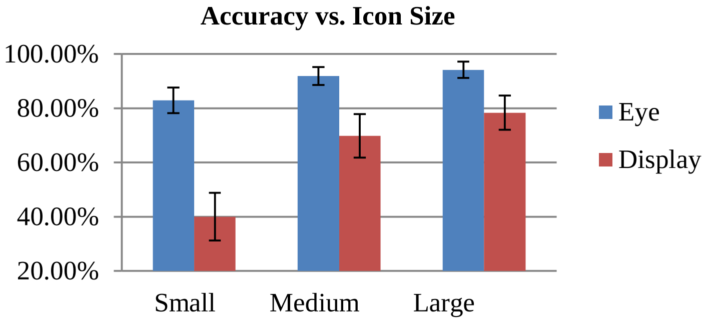


Figure 35 Graph showing accuracy (correctly detected icons) according to icon size and standard deviation**.**

5.2.3.2 **Results**

From the experiments, I was able to evaluate the display in terms of both visual acuity and reaction time in comparison with the naked eye. This let me evaluate how the spatial compression would affect users in an environment that requires peripheral attention. To clearly show significant tendencies regarding acuity, I first plot detection rate of the display versus eye according to icon size and display angle, as shown in Figure 35 and Figure 35. A two way analysis of variance (ANOVA) showed a main effect of device ($F_{(1,9)}$=204.4, P<.01), and a slight interaction of size and angle conditions ($F_{(9,180)}$=2.28, P<.02). Although there is a relatively large difference between eye and display for small icons, the difference decreases as object size increases. For objects over 5.8 degrees of FOV, the difference was only 15.8%. This means that pedestrians would likely notice a peripheral car or bicycle at 10 meters away, but would be less likely to notice objects as distance increases. As shown in Figure 35, there is a relatively consistent difference in error rate for all angles, suggesting that the compression of objects into peripheral space works for objects at any angle, potentially over 180 degrees. It should first be noted that because of camera throughput, processing, and display rendering, there is an inherent delay of approximately 150-180 ms between the time an object appears on the projector and the time it was rendered on the display screen.

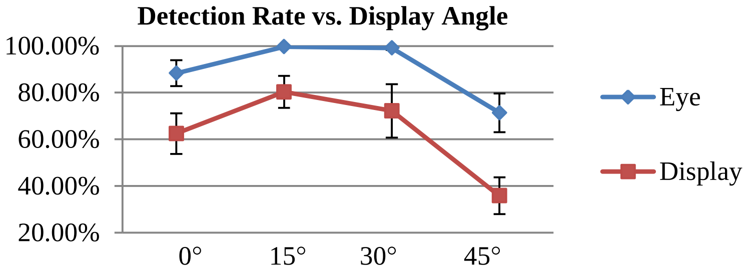


Figure 36 Graph showing average reaction times for all icon sizes for the display (red), for the display minus the inherent delay (green), and for the eye (blue), and standard deviation, with respect to horizontal angle.

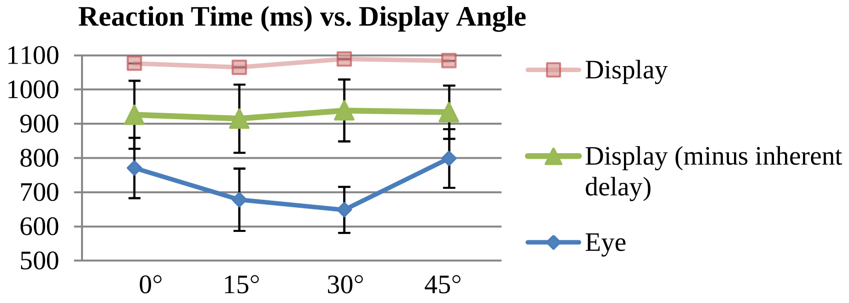


Figure 37 Graph showing averages for correctly detected icons according to horizontal display angle for all icon sizes, with standard deviation.

As such, in Figure 36, average reaction times for the eye, display, and display minus the inherent delay are shown to provide a more objective comparison. A two way ANOVA on delay times also revealed a main effect of device ($F_{(1,9)}$=216.33, P<.01) and slight interaction of size and angle conditions ($F_{(9,180)}$=2.26, P<.01).

Unlike acuity, a very interesting trend occurred with respect to angle and delay. In contrast to the consistent difference in error rates shown in Figure 37, the differences in reaction times between display and eye at 0 degrees and 45 degrees were significantly lower. It is very likely that because different regions of peripheral vision have different sensitivities, delay was higher for the eye at 0 degrees and 45 degrees. This means that placing virtual objects at more central peripheral angles can improve reaction times, which is an important finding for the field of view management. Though other minor tendencies for accuracy were observed, the results discussed above will likely have the biggest impact on future iterations of spatial compression displays. Unexpectedly, no learning effects or improvements in reaction time or detection rate were found for display or eye over time.

5.2.3.3 **Discussion**

Through this experiment, I was able to show that Fisheye Vision enables users to see objects at 180 degrees. With further camera and parameter optimization, this can potentially be expanded past the human visual field. However, based on these results, a number of objects, particularly small objects, will not be noticed with this kind of compression. Part of this difference is probably due to the resolution limitations of the display and camera, so improvements in display technology will likely reduce the disparity between head mounted displays utilizing this method and the human eye. The functionality of the display will likely supersede human vision as technology improves.

In many respects, the compression of the fisheye lenses function like parabolic mirrors on street corners or the curved mirrors attached to many side mirrors on cars nowadays. Although these spatial compression methods provide a better view of the environment, warnings such as “objects are closer than they appear” are often necessary. A similar notice may be beneficial for Fisheye Vision. Also, when testing the 238 degree fisheye lenses, I made an interesting discovery. If both lenses (not virtual, just the lenses) are placed and aligned directly in front of a user’s eyes, the brain can still easily maintain stereoscopy. This means that the brain can stitch together two radially distorted images, which is very interesting physiologically, especially considering the mechanisms behind binocular summation are not yet fully understood (Pardhan and Whitaker, 2000, Wood et al., 1992).

Lastly, studies show that variations in linear scale can cause simulation sickness (Draper et al., 2001), however; the same is not necessarily true for non-linear distortion, such as that of a fisheye lens. This may be even less so when distorted information is only displayed in the periphery. I also conducted several informal tests with the display such as shopping at a convenience store and playing tennis, as shown in C of Figure 32. Initial results indicate that simulation sickness is not a problem, but eye fatigue occurs with prolonged use. As future work, I plan to test fatigue, naturalness, and required mental workload of linear versus radial distortion in a number of concentration intensive outdoor tasks.

5.3 Controlling Vision Augmentations with Eye Tracking

Although I had successfully implemented one type of augmentation, the form factor of the Fisheye Vision display and many of its predecessors were for the most part fixed. It is painstaking to have to set up or use a completely different display every time the user desires a new augmentation.

With that in mind, I decided to build a display that would allow users to easily switch augmentations in real time and control those augmentations with eye movements or gestures. The framework, including hardware and software is called ModulAR, or Modular Augmented Reality, which was designed to improve flexibility and hands-free control of video see-through augmented reality displays and augmentative functionality. To accomplish this goal, I introduced the use of integrated eye tracking for on-demand control of vision augmentations such as optical zoom or field of view expansion. Physical modification of the device’s configuration can be accomplished on the fly using interchangeable camera-lens modules that provide different types of vision enhancements. I implemented and tested functionality for several primary configurations using telescopic and fisheye camera-lens systems, though many other customizations are possible. I also implemented a number of eye-based interactions in order to engage and control the vision augmentations in real time, and explore different methods for merging streams of augmented vision into the user’s normal field of view. In a series of experiments, I conducted an in depth analysis of visual acuity and head and eye movement during search and recognition tasks. Results showed that methods with larger field of view that utilize binary on/off and gradual zoom mechanisms outperform snapshot and sub-windowed methods and that type of eye engagement had little effect on performance.

5.3.1 Introduction

The recent advance of wearable displays has led to a number of new opportunities in the field of Augmented Reality (AR). New applications, research, and commercial products are on the rise, with examples including stereo camera systems for view modification such as Ovrvision and the Oculus Rift, AR binoculars, surgical AR variscopes, night and thermal vision displays, and spatial compression displays (Ardouin et al., 2012, Birkfellner et al., 2002, Oskiper et al., 2013). Many of these video see-through displays allow for augmentations that surpass the abilities of human vision, such as image magnification or modification and vision past the field of view (FOV) of the eye. However, a common characteristic of each of these devices is that they are often fixed to a single display modality or location in the FOV. For example, AR binoculars can only be used to augment distant objects, and combinations of stereo cameras with AR software are just designed to display virtual augmentations in the real world.

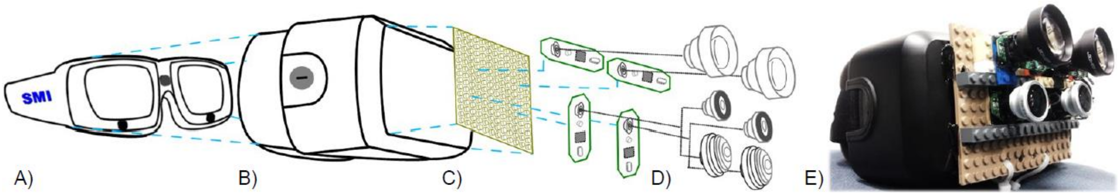


Figure 38 Image showing the general structure of the hardware prototype, including the A) SMI stereoscopic eye-tracker (later integrated directly into the display), B) Oculus Rift DK2 head mounted display, C) modular attachment plate used to interchange various camera-lens modules, D) stereo camera pairs with telescopic, fisheye, and ultrawide vision augmentation lenses, and E) an example of a configuration that allows the user to merge a binocular telescopic video stream into a one-to-one field of view.

Though different functions like optical zoom, peripheral compression, and adaptive visualizations exist, they must often be engaged manually and are rarely implemented in the same device. Even if these functions were available on-demand, questions still remain as to how a user would engage them in a hands-free manner, what kinds of interactions perform best, and how to merge vision augmentations into the FOV in an intuitive and non-disruptive way. Problems such as distorted FOV or amplified lateral translations in the telescopic video streams must also be overcome. As a step towards addressing these questions and issues, I introduce Modular Augmented Reality (ModulAR). Simply put, the goal is to build a video see-through display with configurable hardware augmentations and minimally disruptive software visualizations that can be freely engaged via intuitive eye movements. This contribution includes a hardware prototype, software framework, and a number of methods for interaction and visualization that I have tested and refined through experimentation.

First, I propose a hardware framework that gives users the ability to interchange camera-lens systems (modules), and camera-lens pair locations, as shown in A – E of Figure 38. This allows for wide FOV video see-through configurations such as Ovrvision or the display proposed by Steptoe et al. (Ovrvision, 2015, Steptoe et al. 2014), telescopic functionality similar to AR Binoculars (Oskiper et al., 2013), and stereoscopic spatial compression such as in Fisheye Vision (Orlosky et al., 2014). Physically, this means that users can have on-demand zoom, see-through AR, and FOV expansion capabilities all in a single device. Through integrated stereoscopic eye-tracking, users can engage these functions with eye gestures in real time.

Second, in addition to the hardware framework, I also propose an interactive software framework that facilitates different ways to engage and display the various augmentative functions. For example, one way zoom functionality can be engaged is by squinting, which often occurs when a user wants to see a distant object of interest more clearly. Once an augmentation has been engaged, the framework can also deal with a number of unwanted visual effects caused by certain optical elements. For example, I use head tracking to deal with unnecessary lateral translations from jitter caused by head movements when using telescopic functionality. Also, instead of displaying vision augmentations as a static sub-window, I developed a number of ways to dynamically merge magnified or expanded views into a standard see-through view. These include manipulations of transparency and size and overlaying the sub-window onto a region of the display frustum selected by gaze.

Last, I conduct several iterations of each interaction methodology and test efficacy, usability, and general comfort through experimentation. The experiments are designed to 1) determine general accuracy of the engagement detection algorithm 2) compare visual acuity using standard lenses, telescopic lenses, and a user's natural vision, 3) evaluate eye controlled interactions for engaging vision augmentations, 4) test different visualizations which merge telescopic and one-to-one see-through functionalities into the same FOV, and 5) test the amount of error generated by manual reconfiguration of camera-lens modules.

Next, I will describe prior work and outline how this prototype builds on and contributes to state of the art research. Secondly, I describe the hardware and software frameworks, including a detailed description of each hardware module, the eye tracking and engagement detection process, different types of augmentations and visualizations, and overall interaction methodology. I then describe a series of experiments testing efficacy and intuitiveness of the various interactions and visualizations in visual search and recognition tasks. Results are then discussed in detail, followed by other insights and future directions. Below, the view through the telescopic lenses is referred to as *telescopic view*, and the one-to-one video see-through functionality is referred to as *standard view*.

5.3.2 Prior Work

Though a large amount of work has been done on vision modification and expansion, several displays are particularly relevant to ModulAR. One of these is the variscope developed by Birkfellner et al. to assist surgeons by incorporating a virtual plane into the surgeon's real FOV at the same focal distance in 2002 (Birkfellner et al., 2002). Though the evaluation of the device only measured calibration error, this was a major step in merging a sub-window of real time augmented vision into the user's natural FOV. More recently, Oskiper et al. developed a set of AR binoculars in 2013 (Oskiper et al., 2013), which allow a user to view augmentations through a device with a standard binocular form factor. While all of these devices may be useful for a specific purpose, they have a fixed form factor or FOV, require engagement through physical manipulation of some sort, or obstruct the user's regular FOV in some way. In contrast, the ModulAR prototype can be configured on the fly, can be controlled in a hands-free manner, and introduces software visualizations tailored for merging fields of view.

Gaze has often been proposed as a method for interaction, but simply using *x, y* gaze coordinates for selection or interaction often results in issues like the Midas touch problem.

Consequently, a number of different techniques have been used instead, such as requiring a dwell time to select annotations as in the work by Park et al. (Park et al., 2008). More recent work by Lee et al. proposes the use of half-blinks for interaction by tracking the user's lower eyelid (Lee et al., 2010). Gaze depth has also been proposed as a method for engaging and disengaging augmentations, but it requires a visual target to be present beforehand in order to make the selection (Toyama et al., 2015). Blinking and closedness have also been proposed as a method for monitoring health and drowsiness (Le et al., 2013). However, no study has used an eye-controlled mechanism to engage and manipulate the mode of vision augmentation to date.

In a number of different situations, there are many occasions when users may want to enhance vision in a hands-free manner. Examples include sightseeing, mobile navigation, cycling, or military operations. Whereas many other works focus on the implementation of tracking or rendering virtual objects, I focus on the hands-free engagement of vision augmentations, and propose a number of different ways to switch between vision enhancements and standard modes of operation.

Additionally, I set out to find the best combination of vision augmentation, visualization, and eye tracking interaction to engage vision augmentations in real time. Neither merging live vision augmentation streams with standard views nor eye tracking for engaging this kind of vision augmentation in an immersive display have been tested. Moreover, there is a need to make such a framework flexible, so that it can be easily configurable and replicable for other researchers.

5.3.3 ModulAR System Framework

5.3.3.1 **Hardware**

The hardware setup includes several distinct components: the immersive display, integrated eye tracker to conduct eye tracking and process other eye/face movements, modular board mount, detachable camera modules, and different lens types which are affixed to each camera. A visual overview of the ModulAR hardware framework is shown in A - D of Figure 38.

The hybrid eye-tracker and immersive display I am using is a prototype developed by SMI for the purposes of conducting eye-tracking for virtual reality (VR) and AR applications. The stereoscopic infrared (IR) eye tracker is integrated directly into the housing of the Oculus Rift, as shown on the left of Figure 39. The outer rim of each lens contains 6 embedded IR LEDs to illuminate the user's eyes, and eye tracking cameras are situated behind the lenses. These cameras are not visible to the user so that the eye-tracking process does not interfere with the view through the display. The buses for both eye tracking cameras are connected through the single USB port on the Oculus Rift, so additional cables are not necessary.

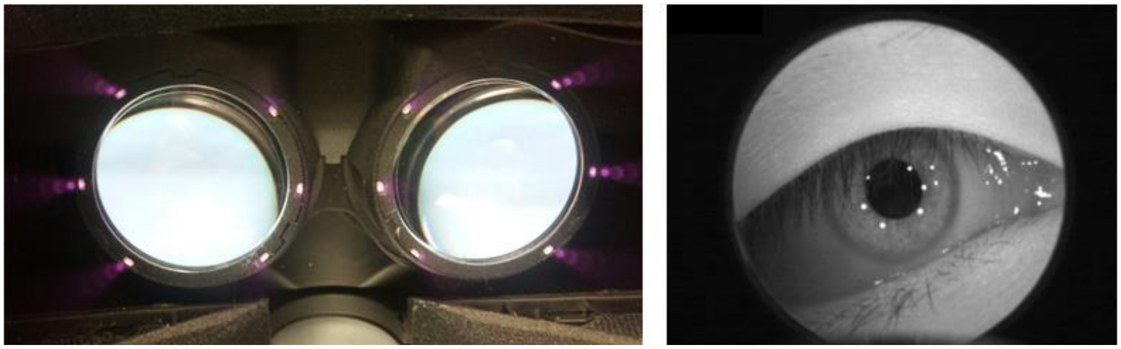


Figure 39 Image of the infrared LEDs that have been integrated directly into the frame of the Oculus Rift (left), and a screenshot taken from one of the eye tracking cameras (right).

Although the specifics of the tracking algorithm are proprietary, eye tracking can still be conducted with approximately 0.5 degrees of angular accuracy despite acquiring images of the eyes through the curvature of the Oculus Rift lenses, as shown on the right of Figure 39. This allows for eye tracking in real time, as well as approximately 90 degrees of horizontal FOV for visualizations.

In order to maximize flexibility of augmentations and still have a display that can be used for practical AR, I made use of LEGO building blocks. While LEGOs are often thought of as children's toys, they function well as a rapid prototyping system, and have been used for designing physical robots, in AR and VR applications, and for modifying camera phones in the past (Irawati et al., 2008, Reshko et al., 2002). Moreover, parts from all LEGO pieces are interchangeable, precision machined, and inexpensive. Because of this machining and flexibility, two affixed LEGO pieces will maintain the same orientation despite numerous detachments and re-attachments. By rigidly affixing camera-lens systems to LEGO pieces, a variety of calibrated camera systems can effectively be swapped and interchanged with relatively little error or recalibration requirement. As a simple example, a stereo camera pair can be calibrated once, removed, and reattached in the same orientation within a matter of seconds, without having to recalibrate in most cases. Exactly how much calibration error occurs during reattachment is described in “Post Test of Calibration Error for Reconfigurations” in the experiments section.

Consequently, I achieve a modular display on which users can quickly swap and modify augmentative configurations. Moreover, sets of camera configurations can easily be swapped and shared between any displays with an attachment plate, meaning other researchers can more quickly and easily share, construct, replicate, and calibrate new setups with a ModulAR device. Other potential configurations are shown in Figure 40. With various camera-lens systems (modules), a number of optical vision augmentations can be achieved. The first module I built was a standard AR camera pair, which was based on systems like Fisheye Vision and the display built by Steptoe et al. (Orlosky et al., 2014, Steptoe et al., 2014).

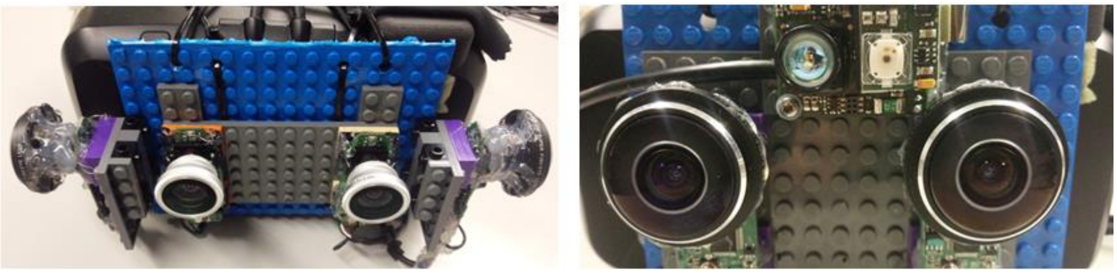


Figure 40 Side-mounted fisheye camera system for viewing peripheral and rear scene information (left), and a setup with wide-angle lenses and time-of-flight camera for hand tracking and real time 3D reconstruction (right).

The cameras for each module are Logitech C310s set at 800x600 resolution, which provide ample control over parameters such as frame rate, exposure, and gain. Some stereo camera rigs for the Oculus Rift use lenses with the same FOV distribution of the Oculus Rift. However, I chose 180 degree wide FOV lenses so that peripheral information can be accessed, as can be seen on the lower part of the right hand sample in Figure 38. To achieve the 1 to 1 AR video see-through display (standard view), I compensate for lens distortion using OpenCV's checkerboard calibration and undistortion libraries. More importantly, by accessing the full undistortion map, additional peripheral information is available, which can be used to compress peripheral views of the fisheye lenses or merge rear-view information visible from side-mounted cameras. A second set of lenses with a 238 degree FOV have also been assembled, calibrated, and tested in previous experiments, as outlined in Fisheye Vision (Orlosky et al., 2014). The experiments showed that peripheral vision expansion can be used to view objects in a 180 degree FOV while still allowing the user an undistorted central field.

Since I had already tested merging of a compressed peripheral view, I chose to implement a telescopic camera-lens system in order to add telescope, scope, or binocular functionality (telescopic view) for experimentation, since merging telescopic video see-through functionality with a stereoscopic standard view has yet to be studied in depth. The telescopic lenses used are 5x optical super telephoto lenses by Locofoto, and each is fixed to one of the cameras at an optimal distance between the telescopic lens inlets and the web camera lenses.

5.3.3.2 **Software Framework**

Software is composed of the eye-tracking framework, interaction detection algorithms, a game world for integrating virtual objects, visualizations of augmented data, and finally computer vision algorithms for calibrating, undistorting, and merging augmented video outputs from different camera-lens modules. A visual representation of the entire software framework is shown in Figure 41. To present a correct view of both the standard AR view and telescopic lenses, calibration must occur prior to runtime, and undistortion of all 4 camera-lens systems must occur in real time. First, to obtain undistorted views of all cameras, I use OpenCV’s undistortion functionality to correct for radial distortion (Bradski et al., 2008).

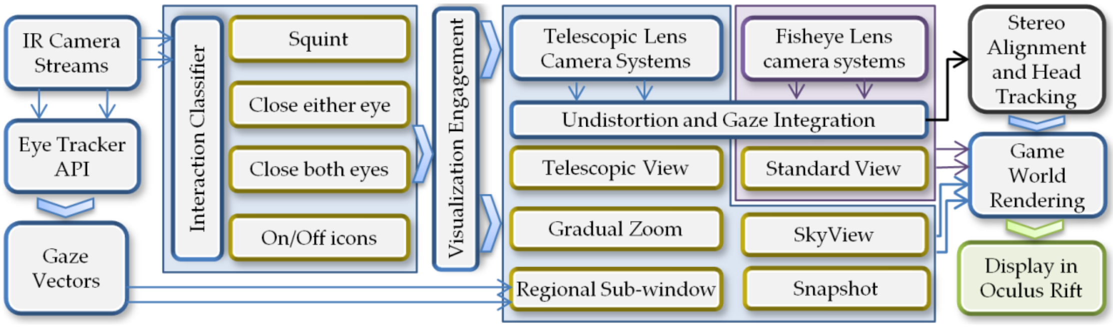


Figure 41 Software flow diagram showing eye tracking, engagement, visualization, rendering and undistortion processes.

Camera images through the fisheye lenses are faced toward a checkerboard to obtain camera parameters. After obtaining radial and tangential distortion coefficients for each lens, a remapping function is then applied in real time to all four video streams to correct lens distortion. The same calibration is carried out for the telescopic lenses. In order to display the camera video streams and render virtual objects, I have merged the video see-through view with the lightweight Java gaming library (lwjgl). This allows for rendering of video streams from all 4 cameras as textures, moving them in the 3D virtual environment as necessary, and also rendering virtual objects on top of the see-through components for AR/MR. The rendering process for these textures has been hand optimized so that the two IR eye tracking cameras, four c310 web camera streams, gaming library (including the barrel undistortion and aberration correction for the Oculus Rift), and framework can all run in real time at over 30fps. Note that cameras should have their own USB bus if possible due to the fact that numerous devices on a single bus can cause initialization and rendering errors on some machines.

For the standard see-through AR view, head tracking is disabled on the web camera textured objects so that the images are left directly in front of the user's eyes. At the same time, the game world is rendered around the web camera objects, which can be seen in the background in F of Figure 43. Due to the time it takes for the last pixel of each 800x600 pixel image to reach the software framework, an inherent delay of approximately 100-150 milliseconds is present in the web cameras, which is greater than the tracking delay for the Oculus Rift sensors. Consequently, I actually slow down head tracking to match the delay of the web camera hardware. In short, the camera images that change as the user's head rotates will be rotating at the exact same rate as the game world, which allows for more realistic rendering of virtual objects.

5.3.3.3 **Eye-tracking for Engaging Vision Augmentations**

To process eye movements, custom eye tracking software developed by SMI for the Oculus Rift is used, which is adapted for use with the framework. In addition to using the *x*, *y* coordinates provided by SMI's API, methods for distinguishing several gaze based interactions are presented and described in detail below.

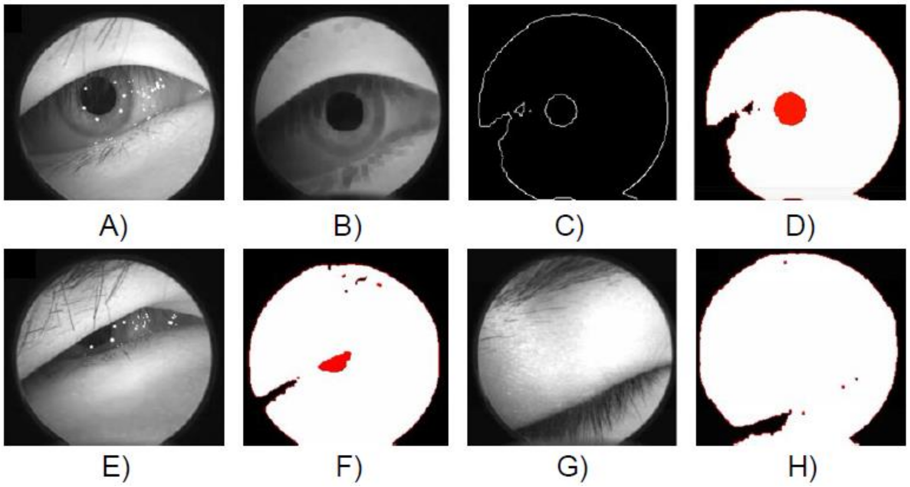


Figure 42 View through the IR cameras showing A) an open eye (raw image), B) with noise removed, C) after applying a binary threshold and canny edge detection, and D) pupil contour detection. On bottom, E) a squinted eye, F) partial pupil that crossed the threshold for squint detection, G) a closed eye, and H) resulting image with no pupil.

Images of the eyes are first received from the IR cameras as shown in A, E, and G of Figure 42. When the eyes are open, respective gaze vectors are provided by the SMI eye tracking API. These gaze vectors are then mapped to 2D screen coordinates in the Oculus Rift, and can be mapped to 3D scene coordinates in the virtual world's coordinate system if desired. For detecting eye activity (open, squint, and close as described in more detail below), I needed to develop a custom algorithm that directly processed raw images of the eye. First, input images are preprocessed by erode/dilate to remove noise, as shown in B of Figure 42. A binary threshold and canny edge detector are then applied to find boundaries as shown in C. Suzuki's contour detection algorithm is then used to find the pupil as shown in D (Suzuki, 1985). Note that this works even when only part of the pupil is visible, which is essential for detecting squints.

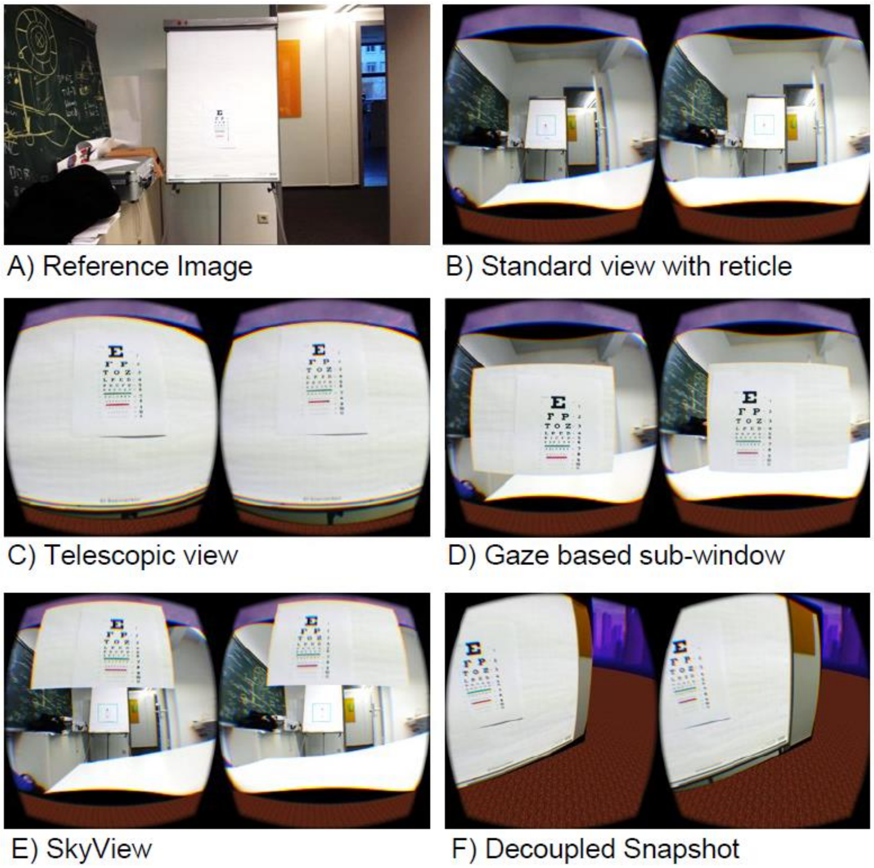
A 'squint' is triggered when a user has partially shut his or her eyes over a certain threshold. Squinting is a common natural movement for someone who is attempting to focus on a distant object. The muscles in the eye contract to change the shape of the cornea, effective exit pupil (aperture) also decreases in size, and the individual's visual acuity changes. Consequently, I thought this would be a natural type of interaction for engaging the telescopic view. One more benefit of squinting for engaging a vision enhancement is that the pupil is always visible. Even if the eye is partially closed, a portion of the pupil must still always be visible in order for light to pass through to the retina. Although an initial calibration is necessary, the squint detection is robust for the purpose of engagement, and accuracy is later described in the experiments section.

Detection is accomplished by finding the largest region outlined by the contour algorithm shown in D and F of Figure 42, which is taken as the pupil region. The area of the pupil within the image is given by the number of pixels P_N shown as the red region in D and F of Figure 42. Since the (estimated) pupil size P_S is also provided by the SMI API, the visible area of the pupil can be estimated by $P_N/P_S = p$. A user-specific threshold t_s for squint detection is then defined. If p is larger than t_s, the current eye state is classified as 'open'. Otherwise, a 'squint' is detected, as shown in E and F of Figure 42. Note that pupil size can change depending on the brightness of the screen, so for robustness, the ratio of visible area P_N/P_S is used instead of the absolute number of pixels in the pupil region. Because a user may sometimes squint unintentionally, the squint engagement is not triggered unless the squint has been detected over a certain number of frames. This number of frames was set to 20 based on initial tests, which is approximately 700ms.

In addition to squinting, a timed full-close (i.e., 'long-close') was also implemented. To engage, the user simply had to close his or her eyes for over 500ms, which prevents false positives from the user's natural blink mechanism. Similar to the squint detection, a ratio threshold t_c for the close engagement is defined. If p is larger than t_c, the activity is classified as open. Otherwise, it is classified as close. Since squinting may be more difficult for some users, it was hypothesized that this timed close might be easier in general.

A "double blink" was also implemented and tested. Double blinks do not often occur naturally, and allow for more consistent perception of the current gaze target. For detection of the double blink, the number of times the eye is closed within 20 frames (≈700ms) is used. If a transition from 'open' to 'close' is detected more than once within this time period, a double blink engagement is triggered. I hypothesized that this method would be slightly more efficient than the timed close, but might result in more fatigue.

Since the primary goal was to test methods for engaging augmentations, the telescopic view was automatically disengaged after two seconds following any engagement. For the experimental task, this would allow users to get a good view of magnified content, and then return to the standard view. Implementing a single eyed close was also considered, which is fairly easy to do algorithmically. However, during initial testing, it was discovered that not everyone can close just one eye at a time, so this method was left out of the experiments. Icon based engagement, i.e., looking at on/off icons located on-screen to engage or disengage augmentations, was also considered. Unfortunately, this type of interaction can result in screen clutter, requires the user to break his or her current line of sight, and may not be natural. Size and screen placement would also add too many variables to the experiment to provide an objective analysis. Vergence based engagement/disengagement would likely be the most natural form of interaction (Toyama et al., 2015), but the user must already have a gaze target in his or her FOV, and might not necessarily want to engage the telescopic view for targets beyond a certain distance.



A) Reference Image

B) Standard view with reticle

C) Telescopic view

D) Gaze based sub-window

E) SkyView

F) Decoupled Snapshot

Figure 43 A) Image showing a picture taken with a camera phone for reference. Next, screenshots taken through the ModulAR device with B) video see-through capability (a square blue reticle is overlaid to indicate the zoom region), C) a view through the telescopic lenses, which can be activated in both binary on/off and gradual zoom fashions, D) a gaze-selected sub-windowed telescopic view, E) the same as C, but without gaze tracking and with an upward shift in y coordinates, and F) a view of a snapshot taken from the telescopic lenses that has been decoupled from the user's view and registered in the game world as a stationary object (the user then moved his head to inspect the lower right corner of the snapshot.)

5.3.3.4 **Visualization of the Telescopic View**

Because the stereoscopic merging of telescopic and standard views into the central FOV is still largely unexplored, I developed and tested a number of visualizations for this purpose as described below. Binary zoom is the most basic type of telescopic merge functionality and simply overlays the telescopic image onto the same FOV as the standard see-through view. In this visualization, a user would see the standard view from B in Figure 43, and the frame following eye engagement would contain the telescopic view shown in C. This method provides the fastest transition from standard to telescopic view, but may also result in difficulty reconverging on the new image immediately after engagement.

Gradual zoom is similar to binary zoom, but the telescopic view is gradually brought into the standard view by increasing both size and transparency of the objects containing the telescopic video streams. Over a period of 500ms, opacity is increased linearly from 0% to 100%, and the telescopic view appears from the center of the reticle shown in B of Figure 43. The final image after 500ms is the same size as in C. The location of the magnified window is adjusted so that the window appears to grow out of the middle of the magnified position, which allows users to more easily judge where the resulting window will appear. For experiments, this reticle was removed to provide a more objective comparison across visualizations. Though the overall transition (500ms) is slower than binary zoom (one frame), this method provides a smoother transition between standard and telescopic views, and was preferred aesthetically during initial tests.

Gaze based sub-regional zoom is defined as the subsection of the telescopic view within the user's gaze at the time of augmentation engagement. When testing binary and gradual zoom, I quickly realized that the region at which the user is looking is not that which is actually magnified since the telescopic lenses have a fixed position relative to the standard view. To solve this problem initially, I added the reticle shown in B of Figure 43, which indicates the region that will be magnified. However, since the user's gaze is available to us through the eye tracker, I can appropriately select a specific sub-window on which to execute the zoom. This is limited to the maximum region of the telescopic lens camera feed, but still works for any gaze points within approximately 30 degrees from the center of the telescopic lens FOV. In the event that the user's gaze has exceeded the telescopic viewing region, the region at the outermost point of the telescopic view is returned.

The implementation for this particular method is more difficult since four coordinate systems must be aligned, including the game world, standard view, telescopic view, and eye-tracking coordinates. In addition to magnifying the region at which the user is looking, the center of the resulting sub-window must also be shifted to the user's current gaze position. Finally, the telescopic image plane is moved slightly closer to the user in 3D space so that they perceive the magnified image as nearer.

Gaze Based Full Digital Sub-regional Zoom is the same as gaze based optical sub-regional zoom, but uses a software magnification of the standard view instead of optical magnification of the telescopic view. This was implemented to address the fact that optical zoom is limited to the region of the telescopic lens. Digital zoom allows us to access any information in the standard view, meaning regions outside the FOV of the telescopic lenses can also be magnified. Due to the fact that resolution of the magnified image is greatly reduced with this view, it was omitted from primary experiments.

SkyView displays the telescopic view as an always-on sub-window above the standard view. In this implementation, the SkyView is always visible, and the user is provided with a reticle, as shown in E of Figure 43. While gaze tracked zoom is useful for engaging a window in central vision, showing the telescopic view in a region in peripheral vision may be more convenient. Although switching of gaze from the reticle to the telescopic view in the periphery takes some time, this window could potentially be left on, removing the need for engagement. However, since I wanted to test engagements, this view was also disengaged for experiments. I hypothesized that this type of view may perform better for searching since the central field is not occluded.

Decoupled Snapshot allows the user to take a snapshot of the current telescopic view, decouple the window, register the resulting image plane in the game world, and inspect the image in detail using head movement as shown in F of Figure 43. This method was designed to eliminate amplified lateral translations due to head movement when the telescopic view is engaged. One example of these translations would be holding a pair of binoculars or telescope, where small head or hand movements result in large, unwanted changes in the user's FOV. This kind of change in movement scale has been shown to cause simulation sickness, and should be reduced as much as possible (Draper, 2001).

Upon engaging this visualization, the current homography of the telescopic image plane is calculated and saved. The Oculus Rift's built in head tracking sensors are then engaged, letting a user temporarily navigate the last frame of the telescopic image with head movements. The snapshot then appears as if it is being held in place (rather than the shaky image often seen through binocular or scope lenses), and the user can then take his or her time to read or examine the image contents with relative ease.

5.3.4 Experiments

A series of user-based experiments were carried out, starting with two short tests to see how much of an improvement was gained by the telescopic lenses and to determine engagement recognition accuracy. A longer primary experiment employing search and recognition tasks was then conducted to test combinations of eye-tracking based engagements and visualizations of the telescopic view. A final test was conducted to determine how much recalibration is necessary after reattaching or reconfiguring camera-lens modules.

In total, 10 individuals (7 male and 3 female) participated, with a mean age of 29.7 (std. dev. 14.5). Participants were seated at the end of a conference table facing a 1920 x 1080 pixel 13 inch (near) and 1600 x 1200 pixel 19 inch (far) monitor for the reading and search tasks as shown in Figure 45. To the left of the monitors was an easel containing the Snellen eye chart shown in Figure 43 to facilitate the basic eyesight test. All user experiments were within-subjects and conducted in controlled lighting.

5.3.4.1 **Test of Engagement Detection Accuracy**

To provide a basic evaluation of the engagement detection algorithm, accuracy of detection of each participant's open, squinting, and closed eye states was evaluated first. Since only three states are classified, open, squinting, and closed, (double blink is dependent on correct classification of the closed state), only the accuracy of separation of squinting and closed (engaged) from the open eye (not-engaged) were evaluated.

Setup To test this, three videos, each approximately 300 frames, of the three eye states were first recorded. Each individual frame was then run through the detection module to check whether it returned the correct engagement label. Before taking the videos, each user was allowed to practice squinting, double blinking, and closing for several minutes. During this time, the threshold parameters t_s and t_c were tuned manually by monitoring the value of p in real-time. The videos were then recorded and processed, and accuracy was calculated by:

$$\text{Accuracy} = (F_{Xcorrect}/F_{Xtotal} + F_{Ycorrect}/F_{Ytotal})/2,$$

where F_{Xtotal} is the total number of frames of activity X, $F_{Xcorrect}$ is the total number of correctly detected frames of activity X, F_{Ytotal} is the total number of frames of activity Y, $F_{Ycorrect}$ is the total number of correctly detected frames of activity Y. Averages of accuracy calculated for the 10 participants were 97.5% (std. dev. 2.1%) for the open vs. close test and 94.0% (std. dev. 3.6%) for open vs. squint test. Since the number of detect frames is taken into account for triggering visualizations, we can expect a low number of false-positives. This showed that we can robustly detect and use each type of eye engagement.

Prior to testing the eye-tracking and visualizations, each participant conducted a visual aptitude test. I used the Snellen eye chart (A4 size) commonly used to evaluate eyesight in optometry, and positioned the chart on an easel at 2 meters (m) away from the participant as shown in A of Figure 43. Each participant then put on the prototype and was asked to read characters from the line they thought they could read clearly. They first read the chart with the standard view, then telescopic view, and finally with the naked eye. Correctly reading more than 50% of the characters in a particular line resulted in success for that line. All participants completed the eyesight test in less than 5 minutes, and then proceeded to the main engagement and visualization tests.

Results Results are summarized on the right of Figure 44, which shows the level to which each user could read for the standard view, telescopic view, and naked eye. A single factor analysis of variance (ANOVA) shows an effect of view ($F_{(2,\ 29)}$ = 25.51, P<.01) on acuity.

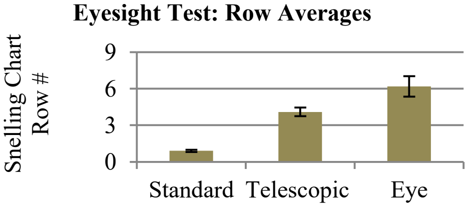


Figure 44 Graph showing results of the eye test, including average acuity with std. error.

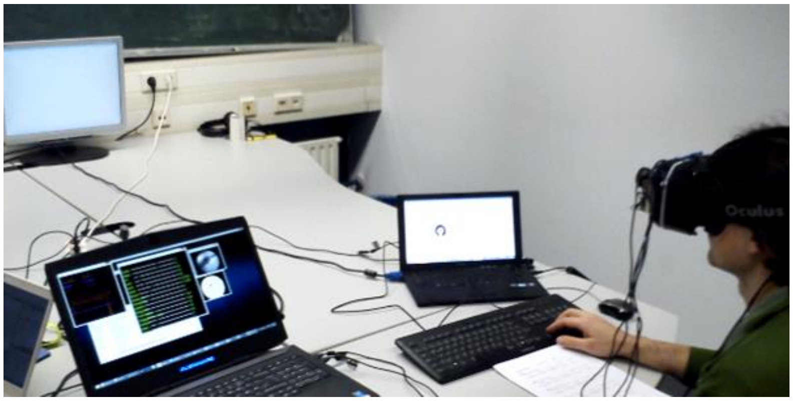


Figure 45 View of the experiment environment showing near and far monitors for displaying Landolt rings.

In general, the human eye still has much higher angular resolution than video-see through functionalities, primarily due to resolution limitations of both camera and display. Letters in line 1 take up 1.81 degrees FOV compared to 0.18 degrees FOV for line 4, so the telescopic view still provides a 10x angular acuity improvement over the standard lenses. The size of each Landolt ring on the far monitor described in primary experiments was designed to be unreadable with the standard view, but readable with the telescopic view for an objective comparison.

5.3.4.2 **Primary Experiments: Evaluation of Eye Engagement Methods and Visualizations for Merging Telescopic View**

Setup To test the system in depth, an experiment was conducted to evaluate each of the three engagements with each of the five visualizations. In short, the task was to determine the orientation of Landolt rings that appeared on the two monitors (near and far), which were positioned at 70cm and 2m away, respectively, as shown in Figure 45. Rings of three different sizes (.12, .15, and .18 degrees FOV on the far monitor, 1.22, 1.80, and 2.37 on the near monitor), and four orientations (up, down, left, and right, randomized and balanced between participants) would appear in random positions on the screen. Rings on the far monitor were set to a size that would not be readable in standard mode (size <= 0.18 degrees FOV), requiring the user to actively switch to the telescopic view. Rings on the near monitor (size >= 1.22 degrees FOV) were readable with the standard view for all participants. This setup was designed to emulate a search, recognize, and record task that might be conducted when navigating, cycling, engaging in military operations, or carrying out tasks in which the user's hands are occupied.

Participants had to 1) engage the telescopic mode to find and view the ring on the far monitor, 2) determine the orientation of the displayed Landolt ring, 3) use the arrow keys on a nearby keyboard to record the ring's orientation, 4) view the next ring on the near monitor with standard view, 5) use the keyboard to record its orientation, and 6) repeat the process for all 24 rings in that trial. Once the participant made a selection using the up, down, left, and right arrow keys, the next ring would appear on the opposite monitor. Appearance of rings was alternated between monitors so that participants were forced to switch between standard and telescopic views for each subsequent ring. Each trial lasted approximately one to two minutes.

Fifteen total trials were carried out to test each of the 3 eye engagement methods with each of 5 visualizations for a 3x5 within subjects design. Engagements included 1) squint, 2) timed close (long close), and 3) double blink. Visualizations included 1) binary, 2) gradual zoom, 3) gaze based regional sub-window (sub-win), 4) skyview, and 5) snapshot. Visualizations were automatically disengaged 2 seconds following engagement. Combinations of these variables were shuffled using a Latin square distribution to alleviate ordering effects. During each of the trials, the following were measured:

- Accuracy (% of correctly selected Landolt ring orientations)
- Trial completion time (time from first to last arrow keystroke)
- Number of times the telescopic view had to be engaged
- Head Movement (pitch and yaw from the Rift's sensors)
- Eye Movement (x, y position from the eye tracker)

These metrics would show which combinations of interactions were better for the engagement and reading tasks. Recorded head and eye movements also enabled analysis of whether users had to physically move their head or eyes more for a particular variable. A subjective questionnaire was also given at the end of each trial with two questions as shown below:

- Was this method intuitive?
- Did you feel fatigued, nauseous, or sick during the task?

After completing each trial, participants removed the display and answered using a 5 point Likert scale ranging from "not at all" to "very much." A one minute break between trials was allowed if desired, and the participant put the display back on to continue to the next trial. Completion time for each trial varied slightly between participants since elapsed time was dependent on how fast the participant selected each Landolt ring orientation, but all participants completed all 15 trials and surveys in less than one hour.

Results The first apparent result was an effect of visualization on accuracy, (i.e., the correctly selected percentage of orientations of Landolt rings). A two factor ANOVA shows a main effect of visualization ($F_{vis(4,135)} = 2.59$, $P < 0.05$) on total accuracy, and paired t-tests show significant differences for binary vs. snapshot and gradual vs. snapshot, $P_{bin\text{-}snap}$ ($t29 = 3.85$, $P < 0.01$) and $P_{grad\text{-}snap}$ ($t29 = 3.85$, $P < 0.01$). Note that all t-tests use Bonferroni correction. The averages according to visualization are shown in Figure 47, showing that binary and gradual zoom visualizations performed the best, and snapshot the worst. For the

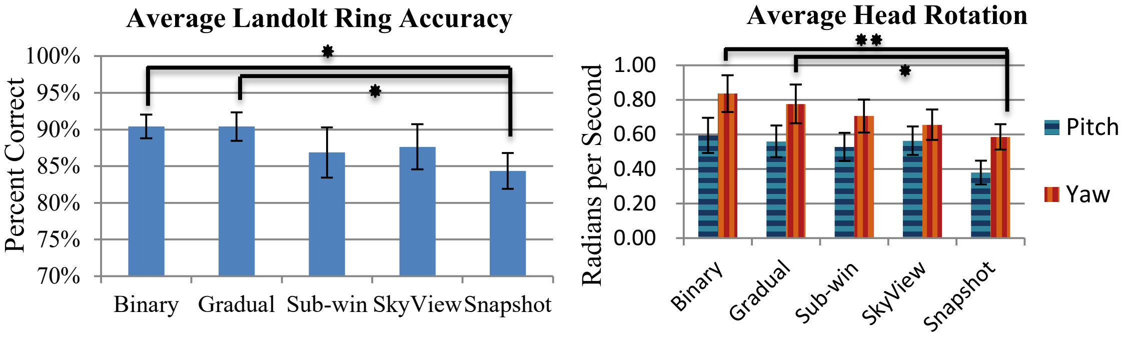


Figure 47 Graph showing accuracy (with std. error) of Landolt ring selections according to visualization (left). Graph showing average head rotation (with std. error) for both pitch (vertical) and yaw (horizontal) according to visualization (right).

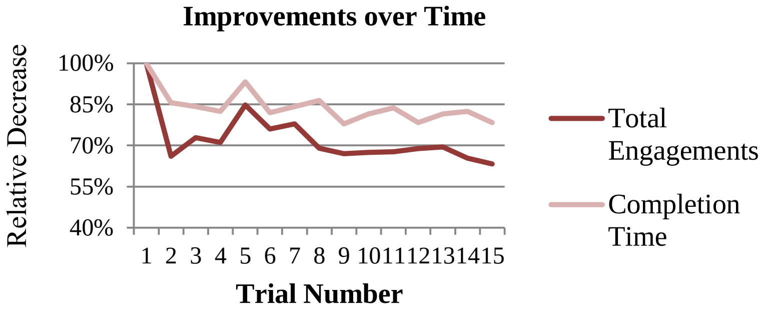


Figure 46 Graph showing reductions over time (in % reduction) for number of total engagements required and average completion time.

regional sub-window and skyview visualizations, it is likely that the lack of improvement in accuracy was due to the decreased window size. The resolution of the image returned by these two methods is exactly the same as binary and gradual, so the size and position of the window are the only factors that would have affected accuracy. One reason that snapshot may have performed poorly is because the image is susceptible to motion blur. Since the user's head is often moving, it is more likely that the snapshot will be slightly blurred when viewed.

In contrast, all other visualizations allow the user to integrate information over several frames. A second reason may be that the ring in the resulting image was not always in the central field of view, whereas participants had more time to center the ring with other visualizations. The next tendency I found was related to visualization and head movement. Movement was calculated by totaling the changes in pitch and yaw for each trial and dividing by the duration of that trial, giving us average angular movement per second in radians per second. A significant effect was found for both pitch (vertical) and yaw (horizontal) movement via ANOVA ($F_{pitch(4,135)} = 2.82$, $P < .05$, and $F_{yaw(4,135)} = 3.09$, $P < .05$) and paired t-tests show significant differences for binary vs. snapshot and gradual vs. snapshot, $P_{bin-pitch}$ (t29 = 4.42, P < 0.01), $P_{bin-yaw}$ (t29 = 3.55, P < 0.01), and $P_{grad-pitch}$ (t29 = 3.85, P < 0.01).

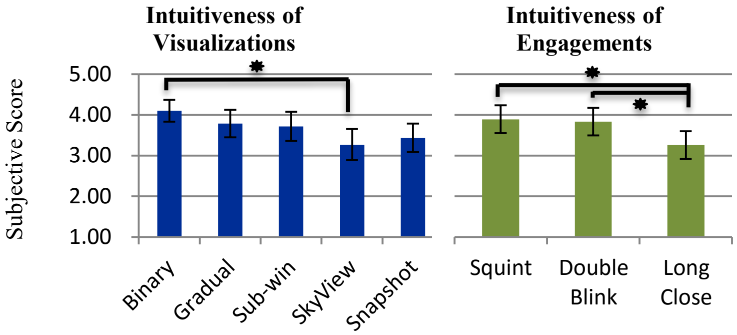


Figure 48 Graphs showing subjective evaluation scores (with std. error) of intuitiveness for visualizations (left) and engagements (right).

Unexpectedly, head movement followed a very different pattern than accuracy as shown in Figure 47. Snapshot actually resulted in the smallest average head movement. No significant effect was found for time to completion, so participants completed trials with less head movement in the same amount of time. This may be because they figured out that keeping their heads still would result in a snapshot with less blur. This should be regarded as a trade-off between the snapshot and binary/gradual visualizations, and an appropriate visualization should be used depending on the desire for higher accuracy or lower head movement.

An effect of session was also observed, showing a gradual decrease in the number of total engagements required per trial and for time to completion. A regression analysis shows a main effect of session on total engagements required ($F_{sessionT(1,14)} = 9.43$, $P < .01$) and on time to completion ($F_{sessionD(1,14)} = 8.19$, $P < .01$), as shown in Figure 46. No effect was found for accuracy of ring selection, meaning that while users can improve their use and interaction with the device over time, acuity will not likely increase. One other interesting result was that no increase in fatigue, nausea, or sickness over time was observed based on question 2 of the survey. ($F_{session\text{-}fns(1,14)} = .28$, $P = .54$). Though the experiment only lasted 45 minutes, this suggests that moderate use of the device may not significantly increase simulation sickness.

A Kruskal Wallis Test on the scores of the subjective survey revealed a main effect of both visualization and engagement on the first question regarding intuitiveness as shown in Figure 48. Results showed that long close was perceived as significantly worse that squint or double blink, ($H_{squint\text{-}close}$=8.087, p< .01, $H_{double\text{-}close}$=6.594, p<.05). Several participants actually mentioned at the end of the experiment that it was easy to lose their place after engaging with long close. Another reason for this is likely that with double blink and squint, the user's perception of the real world is more consistent, and persistence of vision is higher.

Regarding visualizations, SkyView was perceived as less intuitive than binary ($H_{bin\text{-}sky}$=7.32, p<.01). Several participants mentioned that they didn't like having to look up at the skyview visualization after engagement. At first, I thought that this would definitely result in more eye movement for skyview, but no significant effect was found. It is more likely that the skyview was located too far above the user's central vision, meaning that they sometimes had to move their eyes out of the central field of vision. Participants may not always have been able to compensate for the difference with head movement, sometimes requiring unnatural (not necessarily more) vertical eye movement.

5.3.4.3 Post-test of Calibration Error for Reconfigurations

Setup Finally, I conducted a post-test to measure exactly how much calibration error (i.e., drift) would be incurred by removing and re-attaching modules. To measure this, I first rigidly affixed a large 6x12 unit LEGO base piece to a table surface to prevent movement. I then took a standard lens module, which takes up 3x6 units on the base block, and snugly attached it to the base piece. Approximately centered in the FOV of the lens was a small black square marker on a white background, which was to track deviations in placement. I then saved a single frame from the camera-lens video stream to file by pressing space on a keyboard. The module was then detached, and this entire process was repeated 20 times. This simulated 20 "re-attachments" of a camera module to a ModulAR attachment plate. The same experiment was repeated for a telescopic lens module.

Results To calculate drift, I first ran an algorithm to find the centroid of the black marker on each image. Since the marker took up more than one pixel on each frame, I could calculate the center of the marker with sub-pixel accuracy. The *x*, *y* deviations in camera positions that were generated for each frame are plotted in Figure 49, which shows how far in pixels the camera drifted over the 20 re-attachments. Note that these are relative positions of the centroid, not absolute position on the camera image. Drift is calculated as the standard deviation of all 20 re-attachments, plus the first attachment. For the standard lenses, drift was only 0.13 pixels horizontal and 0.28 pixels vertical on the 800 x 600 pixel image. This is unlikely to result in any change in perception for the user. In a single instance, the deviation exceeded one pixel, which is still likely unnoticeable. The telescopic lenses exhibited a significantly larger error, as expected due to amplified misalignments from image magnification. Drift was 0.32 pixels horizontal and 1.63 pixels vertical. In this case, a small recalibration (or re-affixing) may sometimes be necessary to compensate for vertical misalignment. Vertical misalignments are greater due to the fact that the block containing each module is 6x3 units, making it more stable horizontally. This misalignment could potentially be reduced by using a larger LEGO block with more vertical surface area, or more careful placement by the user. Note that this would not be necessary if two camera modules were rigidly fixed to a single piece as a set instead of each camera being attached separately.

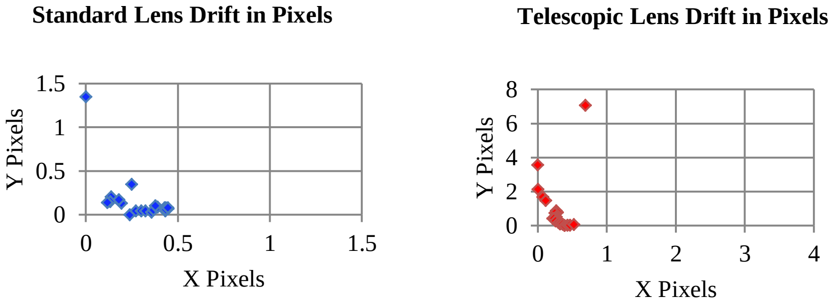


Figure 49 Plots showing how far (in pixels, with sub-pixel accuracy) lenses drifted over the 20 re-attachments for standard (left) and telescopic (right) views.

5.3.5 Discussion

For the experimental task, binary and gradual zoom had both the best accuracy and were the most preferable based on user comments and the survey. However, it is clear that other methods of visualization have advantages for more specific tasks.

For example, someone who might need to read a street sign with large lettering may benefit from the snapshot view. One of the advantages of skyview and sub-window modes is that the user can perceive the normal view and the zoomed view at the same time. Certain tasks such as hunting or surveillance might benefit greatly from seeing two views at the same time, so sub-window visualizations could potentially outperform others. A cyclist may benefit from two sub-windows on either side of the display that function like side-mirrors. Further testing of these sub-window strategies with different window sizes and locations or with an included reticle deserves consideration.

One big limitation of the display is the FOV of the telescopic lens (approximately 30 degrees), so the regional sub-window zoom is also limited. This likely prevented the sub-windowed zoom from being more effective when combined with the eye tracker. The thought was that the user would be able to select the region to zoom more flexibly, thus resulting in fewer head movements. To solve this problem, I could potentially use a motorized telescopic lens which has an adjustable FOV and can change its physical orientation in real-time to match the full FOV of the standard view.

One other benefit of the system is that since the LEGO block positions are fixed, calibrations can be saved and loaded easily as long as modules are affixed to the same location on the base board. Other researchers can then easily replicate a setup as long as they know the x and y coordinates and size of each module. The current biggest challenge is probably the weight of each camera-lens module. While the LEGO blocks are light, the ultrawide and telescopic lenses are relatively heavy (25 and 31 grams each, respectively), and the difference is noticeable. I could potentially solve this problem by using lightweight metals or carbon-fiber shells for the modules.

A next step is to have a completely self-calibrating infrastructure that automatically calibrates regardless of where camera modules are placed. The software framework would also account for various lens modules and adjust the incoming video streams for optimal viewing. One potential way of accomplishing this is by installing miniature sensors into both the attachment plate and modules so that the device can detect both the type of camera and placement/orientation on the board.

5.4 Summary

In this chapter, I introduce the Fisheye Vision and ModulAR displays, which encompass a flexible hardware and software framework designed to improve usability, configurability, and hands-free control of video see-through vision augmentation functionality. After building a number of view expansion prototypes, I then incorporate an eye tracking system that has been integrated directly into the immersive display’s frame to allow users to engage various vision augmentations in real time. A variety of visualizations designed to merge different fields of view were then developed, refined, and tested, revealing many interesting performance tendencies and generating significant qualitative feedback. The method for classifying eye movements for engagement proves to be robust, and I find that re-affixing or re-configuring modules can be accomplished with minimal re-calibration.

CHAPTER 6

Commercial and Interdisciplinary Applications

While the methods proposed in the previous chapters provide a means for improving safety and usability, it was also essential to develop and test AR devices for a number of more specific applications.

6.1 Introduction

For testing in everyday environments, I developed applications and frameworks for a different number of situations where the technology might be useful. This chapter is laid out as follows. I first describe the use of AR and offline localization for navigation. I then introduce a mobile text input called the Torso Keyboard that allows for touch typing while walking. This is followed by a general framework that outlines the use of AR for memory assistance. Finally, I introduce a simulator for monitoring of individuals with cognitive impairments.

6.2 Augmented Reality Navigation and Localization

Indoor navigation in emergency scenarios poses a challenge to evacuation and emergency support, especially for injured or physically encumbered individuals. Navigation systems must be lightweight, easy to use, and provide robust localization and accurate navigation instructions in adverse conditions. To address this challenge, I propose a combination of magnetic tracking and optical character recognition (OCR) in order to provide more robust indoor localization. In contrast to typical wireless or sensor based localization, this fused system can be used in low-lighting conditions, smoke, and areas without power or wireless connectivity. Eye gaze tracking is also used to improve time to localization and accuracy of the OCR algorithm. Once localized, navigation instructions are transmitted directly into the user's immediate field of view via head mounted display (HMD). Additionally, setting up the system is simple and can be done with minimal calibration, requiring only a walk-through of the environment and numerical annotation of a 2D area map. Evaluations of the magnetic and OCR systems are conducted to evaluate feasibility for use in the fused framework.

6.2.1 Introduction

In an emergency, evacuees and rescue teams are faced with a number of challenges when navigating a building or indoor environment. Unfamiliar building layouts, smoke, the absence of lighting, disorientation, or a combination of factors can often prevent an individual from completing navigation tasks in a timely manner, resulting in the need for additional rescue operations or increased risk to the individual. Due to the recent development of smartphones and other sensing systems, researchers have begun build new ad hoc solutions for localization and navigation of indoor environments. Localization of outdoor environments are typically achieved by a combination of sensors such as GPS and compass, but these have limited functionality or usefulness when in an enclosed area, making other means necessary indoors.

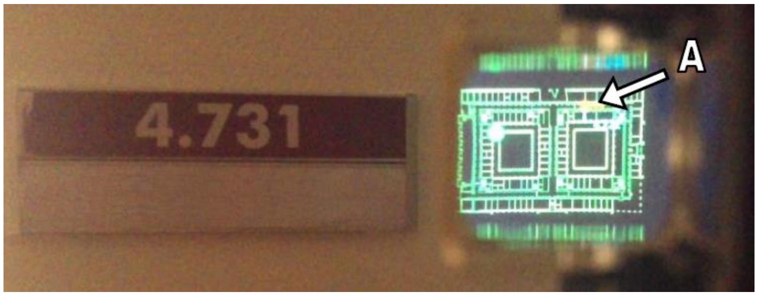


Figure 50 View through HMD screen showing localization (A) on a 2D floor map. The OCR algorithm recognizes door numbers to determine position.

As such, other types of methods such as sonar and network localization have been implemented with some success (Fischer et al., 2008, Inoue et al., 2008, Pan et al., 2006). However, many of these methods depend on consistent network access or detailed 3D models of the intended environment for navigation.

With the limitations of these methods in mind, I set out to develop a lightweight system that can achieve indoor localization despite loss of power or impaired vision due to smoke or dim lighting. After considering numerous possibilities, I chose a combination of optical character recognition (OCR) and magnetic tracking to implement the localization and navigation algorithms. Simply speaking, OCR is used to recognize text in a user's environment when visual data is available, namely room numbers, and determine a relative position on a 2D floor map of the building. This allows for determination of location without a complex model of the environment and despite sudden changes to the scene. Magnetic tracking via tablet is used when lighting conditions such as darkness or smoke do not allow for computer-vision based localization. Additionally, OCR localization and magnetic tracking can be used interchangeably to compensate for changing environmental conditions, and can be used simultaneously by choosing whichever system has higher confidence. Results of two pilot experiments to determine accuracy for the magnetic and OCR systems are also presented. The magnetic system is tested on a variety of data, including localization estimates for an individual in a wheelchair. The OCR system is then tested in different lighting conditions, including nighttime, daytime, and simulated smoke. From this data, I provide an estimate of how the fused system would improve tracking in an actual emergency scenario.

Lastly, navigation information is presented to the user through an HMD, which allows for hands-free operation. An image through the HMD viewing screen showing a user's position localized from a doorplate is shown in Figure 50. An injured person, firefighter that must use rescue tools, or physically handicapped individual can navigate without the use of his or her hands using the system, which is not true for most localization methods that utilize a hand-held device. This intelligent fusion of methods and hardware gives us a number of advantages over other systems in terms of usability, robustness, and simplicity of implementation.

6.2.2 Prior Work

Related research typically falls into two categories or some combination thereof. These include 1) Methods that can be used to localize an individual indoors using a network or other sensors, and 2) navigation algorithms or strategies and studies regarding navigation tasks in emergency scenarios. The remainder of research tends to focus on training, virtually submersive environments, or specialized localization or navigation methods for other scenarios.

One of the cornerstones of a good indoor navigation system is the ability to localize the user both consistently and accurately, especially in an emergency situation. There are a variety of methods available for indoor localization, one of which is triangulation based on wireless signals (Inoue et al., 2008, Rueppel et al., 2008, Tseng et al., 2006). Since the locations of wireless routers typically do not change in the short term, the position and signal strength of wireless beacons can be used in the same way satellites are used in GPS systems. A hand-held device is then used to calculate position based on the relative signal strengths of each beacon. These methods provide accurate localization, but require advanced registration of the position of each wireless beacon and cannot be used if wireless networks become non-functional, making them more difficult or impossible to use in emergencies.

A second set of localization methods includes sensors that are integrated into a handheld device or other ad-hoc networks. One sensor-based method by Fischer et al. is implemented with foot-mounted inertial sensors and ultrasound beacons (Fischer et al., 2008). Again, while accurate localization can be achieved, advanced setup of numerous sensing systems is not feasible for all indoor environments, especially large buildings. Methods for generating indoor maps were developed by Xuan et al., who used both magnetometer and accelerometer data to generate maps for later use with navigation (Xuan et al., 2010). Later, a more flexible, self-contained system based on magnetometers, accelerometers, and optical flow was developed for smartphones in 2012 by Bitsch Link et al. (Bitsch Link et al., 2012). Using this system, an indoor map can be produced and navigated by calculating the speed of a user and monitoring changes in the magnetic signature of a building.

The magnetic tracking I use is relatively similar to these methods, but is improved upon in several ways, such as utilizing OCR to correct a user’s position when localization cannot be achieved from the magnetometer alone. I also discuss several methods for dealing with shifts in the magnetic field or erroneous sensor data which may differ from original mapping data.

The other focus area for this kind of system is on the navigation methods, rather than localization. One such method is presented by Klann et al., which also uses an HMD to present navigation data to a user (Klann et al., 2007). In this study, firefighters navigated paths in a building by placing sensor beacons along their travel path. Though firefighters were able to accurately navigate a set path, the system only provides position data where sensors have been placed. Other navigation algorithms have been developed assuming a previous sensor network is in place, such as that by Tseng et al. This method can provide navigation paths to multiple exits or emergency events, also utilizing a “hazardous region” concept which allows selection of a safest travel path (Tseng et al., 2006). Mirza et al. utilizes a similar approach, but for more confined spaces such as homes or smaller buildings. The navigation method takes into account the position of objects for navigation, and assumes a previously existing map of the environment exists that includes locations of objects as well as frequently visited locations. The remainder of research is related to the higher level design of navigation or training systems or studies on more general aspects of navigation.

The research discussed above has paved the way for current indoor localization and navigation strategies, but has also uncovered a number of new problems. Some of the most important challenges include simplicity of setup and robustness to environmental changes, both of which are especially important in emergency scenarios. To address these concerns, I propose a system that excels in speed of information presentation, improved localization accuracy during a loss of power or network connectivity, and hands-free usage. I accomplish this by: 1) using magnetic localization in darkness or smoke-filled environments, 2) creating an OCR based system that can be used when magnetic tracking fails due to unforeseen changes or inconsistencies with the trained database, and 3) outlining a simple framework in which localization can occur with minimal setup. Additionally, I provide general methodology for the improvement of localization in dynamic environments, and provide users with immediate visual feedback. Since the system is hands free, injured people, handicapped individuals, or rescuers that need two hands for other tasks can use the system uninhibited.

6.2.3 Localization Framework

The system needs to be as flexible as possible, so the magnetic and OCR components can work both independently of each other and cooperatively depending on context. To outline how the system works, I will first describe each of the individual components in detail, and then specify how the methods can be fused. In order to accomplish magnetic localization, I first attempted to use an already existing system called IndoorAtlas, which allows users to create and navigate indoor maps (Haverinen et al., 2009). After several tests, I quickly found that this system is prone to error, and does not give direct access to raw magnetic data. The erroneous localization was perhaps due to shifts in magnetic field or sensor data, so I then decided to come up with a custom magnetic tracking method to solve these problems. When designing the software, I had several goals in mind, including low time to localization, robustness to shifts in magnetic field, and easy integration with other frameworks, such as our OCR based localization.

The first successful method found was to use template matching of a 5 second window of real-time data with a pre-recorded database of magnetic signatures over time. A pilot experiment to test accuracy was conducted in a series of hallways, a total distance of 280 meters.

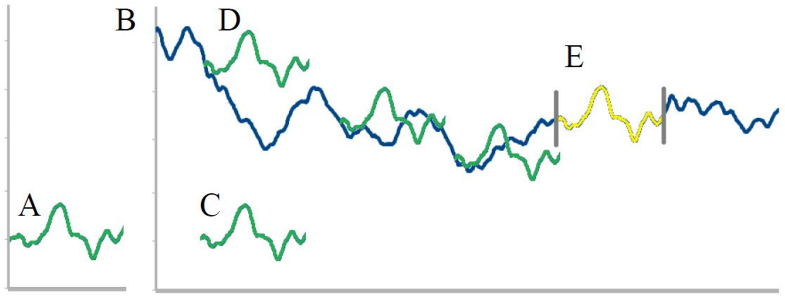


Figure 51 Visual representation of the shift compensation algorithm, including: A) Test data with a significant shift (green). B) Database (blue). C) Bad comparison if no correction for shift. D) Series of comparisons with shift compensation. E) Correct database segment found (yellow).

I found that this method produces very high accuracy (above 90%) assuming the user is walking at a constant pace, resembling the results of other systems tested using robots (Navarro et al., 2009). Note that the results of testing all of the estimation methods are discussed in the experiments section. Unfortunately, humans, especially rescuers or evacuees, do not often move in uniform patterns in emergencies. I conducted several initial tests with both the IndoorAtlas application and the template matching algorithm, finding that accuracy decreases greatly when data is recorded at a different speed than the database recording. Some previous methods attempt to solve this problem using optical flow or accelerometers (Bitsch Link et al., 2012, Steed et al., 2013), but these solutions often require additional sensors, which are not always available and prone to other error.

I succeeded at solving some of these problems algorithmically as described below. There are typically two types of shifts in the magnetic field that are of concern. One type of shift is due to sensors, which can exhibit a gradual shift over several hours, or shifts that are sudden, but less frequent (Steed et al., 2012). I observed several of these sudden changes between the times I conducted initial tests and experiments. The second kind of change is due to equipment or gear a user may be carrying, which I observed when trying to estimate position with a user in a wheelchair. Luckily, none of these kinds of shifts tend to change the general shape of the magnetic field over short windows of time, so simple heuristics can be used to compare against the database with relatively good results. I normalize by subtracting the first value of the comparison window to the first value in the database for every point in time as shown in Figure 51. The normalized input signal is compared using a sliding window algorithm where each window of new data is normalized. I estimate the l1 norm of the difference between each window and the input signal and choose the window in the database with the lowest difference.

Template matching provides some robustness to slight velocity changes, but unless a user starts moving at a constant speed again, magnetic localization becomes inaccurate.

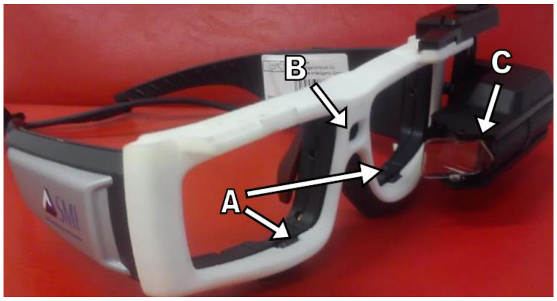


Figure 52 System setup showing A) inward facing cameras, B) outward facing camera, and C) HMD.

I also implemented a method that replaces outliers with previous values that are believed to be accurate, but radical variations in speed can still prevent accurate magnetic tracking. One could try to solve this problem algorithmically using additional sensors, however, the fused approach already provides a partial solution since users can localize with OCR regardless of speed. OCR has traditionally been used for applications like digitization of documents, handwriting recognition, and mail sorting. To my knowledge, this is the first attempt at using OCR to localize a user indoors and in an emergency environment. The idea is to annotate a 2D area map in the same way I connect magnetic signatures to positions on the same map. For example, if a user gazed at the door plate of room 103, and that number was recognized by the OCR, I would then update the user's location on the HMD screen. The calibrated camera system and HMD are shown in Figure 52.

Recognition of characters typically works by some method of pattern or feature matching to find an optimal character candidate out of a set of fonts. Pre-processing and post-processing modifications to the OCR are made in order to improve its efficiency and accuracy. Prior to applying OCR, a gaze-tracking apparatus is calibrated to an outward facing camera. Since the user's gaze relative to the environment is known, we know where the user is looking within several degrees of accuracy. In order to improve efficiency, gaze is restricted to the OCR search window, which improves the time to recognition, and consequently time to localization. The outward facing camera is 1280 by 960 pixels in resolution, and the search window is reduced to 320 by 150 pixels, reducing the number of pixels necessary for search by about 96%. If multiple words are recognized, the closest word to the user's gaze is selected. This also reduces the chances of recognizing some other environmental text as a false positive. The second modification takes place once a raw OCR prediction has occurred. Since there are a limited number of rooms to choose from, a dictionary search is conducted based on the rooms in the database, and only display a position once the OCR has correctly recognized a full room number. The position is then immediately updated on the HMD viewing screen, showing the user his or her current location on the map.

In general, based on the experimental data, taking the localization estimate with the highest confidence will produce the best results. However, I divided possible system usage into three categories and came up with fusion strategies for each. The first is when someone simply needs to reach a given destination. During this time, it is easy for them to travel at constant speed, so the system would use magnetic tracking for a majority of the time, and OCR only when magnetic fails. The second situation includes search and rescue tasks, where someone may have to stop or slow down every so often to listen for survivors or prepare equipment. In this case, magnetic information should be used only when moving between rooms. OCR should be used to confirm position when a user has become disoriented or when a search task has been completed. The last situation is for invasive firefighting or rescue operations. Here, the user will constantly be stopping and changing velocity, so OCR should be used predominantly.

6.2.4 Pilot Experiments

To provide a basic evaluation of the magnetic and OCR components, several experiments were carried out that represent different uses of the system. An initial experiment tests the accuracy of the magnetic tracking at constant speeds and with shifts in the magnetic field. A secondary experiment tests the accuracy of the OCR algorithm in various lighting conditions. I then discuss the feasibility of fusing the two methods based on these results.

6.2.4.1 **Experiment 1: Magnetic Tracker Testing**

The first pilot experiment was designed to test the effectiveness of template matching with a few different types of data, as well as to test the shift compensation and outlier replacement approaches.

Setup I first selected a set of hallways to test against and used an Asus touch screen tablet to obtain data. As the experimenter walked through each hallway, he maintained a constant speed, and recorded checkpoints along the way using a button on the tablet. As he walked, the tablet recorded the current time and *x*, *y*, and *z* components from the magnetic field at approximately 50 millisecond intervals.

The total walking distance used to record the database was 280 meters on a path spanning 7 adjacent hallways. The recorded data was used as a ground truth against which to test localization estimates.

Results To show accuracy, various system estimates versus a ground truth are shown in Figure 53. Additional data to compare against the database was taken by an experimenter at constant speed and also by an individual in a wheelchair. Plot A shows the ground truth, so if the estimate for position p is accurate, it will lie on the same point on plot A at time t.

Table 7 Accuracy estimates

Data	Accuracy of Estimates Using Template Matching			
	Source Data	**Compensation for Shift**	**Outlier replacement**	**Accuracy**
B	Constant Speed	No	No	97.62%
C	Wheelchair	No	No	13.94%
D	Wheelchair	Yes	No	60.08%
E	Wheelchair	Yes	Yes	68.30%

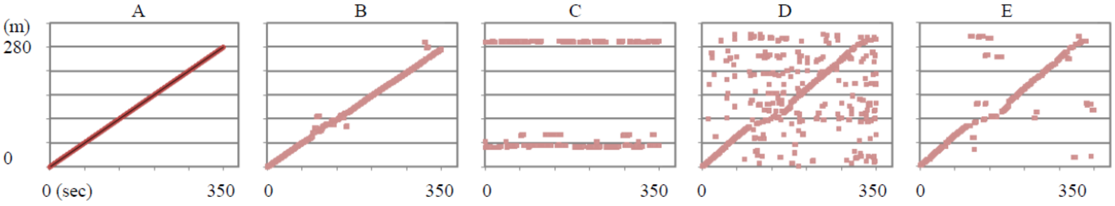


Figure 53 System estimations for user position while traversing a set path. In other words, plots show position (vertical) during a certain time (horizontal). Plots are A) ground truth (actual position), B) constant speed, C) speed is constant but there is a shift in magnetic field, D) the same as C, but with the shift compensation algorithm, E) the same as C, but with both shift compensation and outlier replacement applied.

This system gives an estimate for every position along the path in each plot. Plot B shows system estimates using a window of data taken from a user walking along the same path at the same speed at which the database was taken, without any shifts in magnetic field. Plot C was estimated using data from the handicapped individual in the wheelchair, resulting in an overall shift in the magnetic field. Simple template matching (C) consequently failed. Plot D uses the same wheelchair data, but using the algorithm that compensates for shifts as shown in Figure 51. As shown, predictions are closer to ground truth, but many outliers still exist. I then replaced outliers with a previous value that has a low standard deviation from other previous values, resulting in the plot in E. Though this still cannot achieve perfect accuracy, the results would still be useful for general localization despite shifts in the magnetic field.

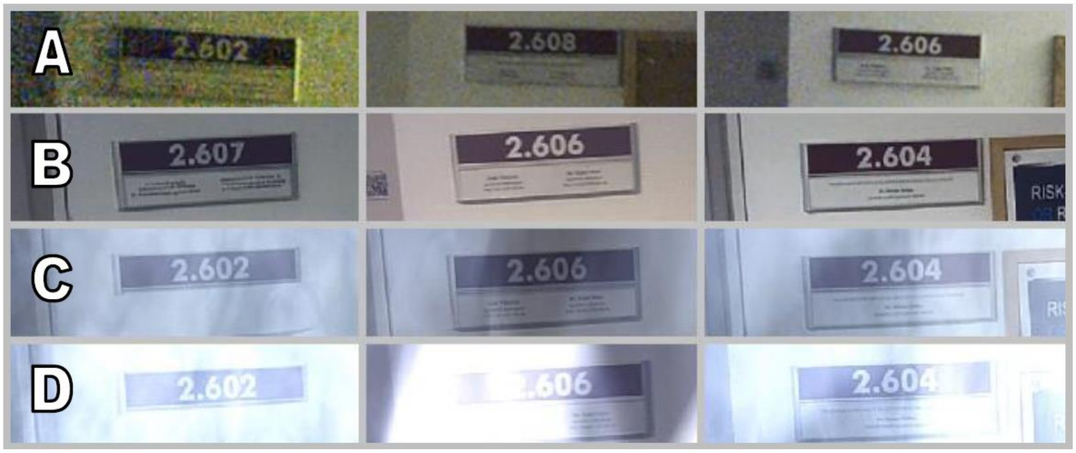


Figure 54 Door plates in various conditions. Row A is in dim emergency lighting. Row B is in standard hallway lighting. Row C is in smoke, and Row D is in heavy smoke. Note that camera exposure automatically adjusted for lighting conditions.

Overall accuracy for each plot is shown in

Table 7. For calculations, a point was considered accurate if it was within approximately 5 meters of the ground truth position along the 280 meter hallway.

6.2.4.2 **Experiment 2: Accuracy of OCR in Adverse Conditions**

In principle, the OCR module can recognize text even in an adverse environment, assuming they are printed with ordinary fonts. However, the user may need to change his or her perspective by changing the distance or the angle to the text, which means that the text may not be recognized quickly. Several other factors that may affect recognition speed, including illumination changes, perspective changes, interference from smoke, and blur. The second experiment was designed to test the accuracy of OCR in an emergency simulation.

Setup To do so, I wanted to test the ability of the algorithm to recognize a doorplate in various lighting and smoke conditions. To measure this, a user traversed a single 40 meter corridor with 11 distinct doorplates and gazed at each door plate while walking. He did this in each of several lighting conditions as shown in Figure 54, and the system automatically recorded whether or not each doorplate was recognized. Video of the hallway was recorded as well, and the OCR was tested on simulated smoke. Three doorplates for each condition are shown in . This shows us how long it might take someone, an en-route firefighter for example, to localize using only the OCR while moving and with limited visibility.

Results In daytime lighting and light smoke conditions, 6 of 11 plates were recognized. In darkness, only 2 of 11 plates were recognized and only 3 of 11 in heavy smoke. Given these results, a user would have to rely more heavily on magnetic tracking in darkness and as smoke increases.

Through the experiments, I reveal the need to compensate for magnetic shifts. When developing any magnetic tracking system, normalization methods should be considered as a potential solution. I also observed problems with tracking due to changes in speed. This problem could potentially be solved with dynamic time warping algorithms, such as those used for recognizing speech patterns in automated phone systems. Other complementary means for example other signals such as accelerometer and gyroscope or classical signal filtering methods such as nonlinear low-pass, high-pass, and band-pass, Kalman, and particle filters and recent signal processing methods could considerably improve performance and compensate for variable speeds (Szabo et al., 2011). OCR is more accurate in regular lighting and light smoke, so reliance on OCR should be used based on context. Given the simple nature of the pilot tests, further testing is needed to evaluate navigation tasks in real time by emergency staff or evacuees, which I plan to conduct as future work.

6.3 Text Input Prototype

Another important factor for the safety of wearable systems is the ability to input text in a manner that does not detract from a user’s typical field of view while walking. To address this problem, I construct and present analysis of a torso wearable text input device. When used in conjunction with a head mounted display (HMD), it allows touch typists to enter text while sitting, standing or walking, and a user can look straight ahead during use. The layout of the keys is similar to that of a QWERTY keyboard, but the device is separated into two halves so that the user can attach each half to the respective side of a suit or vest. It is also flexible and conforms to the shape of the user's body. In the experiments testing input speed, a total of 7 participants conducted various typing tasks, and reached an average speed of 30.1 words per minute (WPM) in a single 45 minute session. The typists were from a variety of ethnic backgrounds and were placed in non-ideal conditions to test robustness.

6.3.1 Introduction

Office workers in many countries sit for long periods every day while editing documents or writing e-mails. If these tasks could be completed in a safe manner while walking or standing, users may be able to reduce stress and lead a healthier lifestyle (Edelson et al., 1989).

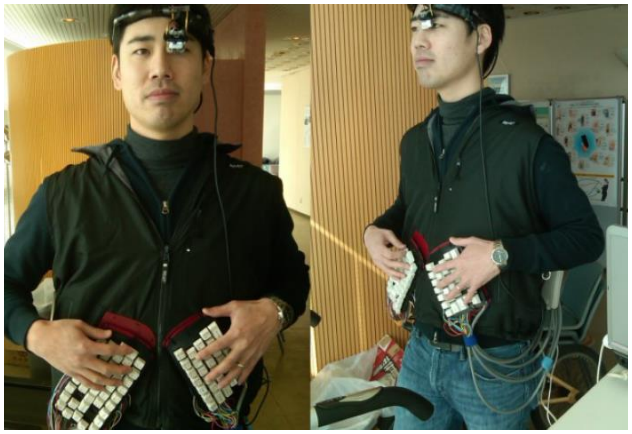


Figure 55 Wearable keyboard prototype used while standing (left) and walking (right).

Also, commuters on buses, trains, and other public means of transportation often use phones for e-mail, which typically have a small screen and relatively low text input speed when compared with other input methods like full QWERTY keyboards. In order to more efficiently enter text while standing or walking, I set out to construct a device that requires minimal learning and enables high typing speeds in situations such as walking to work, commuting by train, or standing in line. To accomplish this, I prototyped a keyboard that can be worn on a vest, shirt, suit or jacket as shown in Figure 55.

I then explore the use of this torso wearable keyboard in conjunction with an HMD for mobile text entry. Though the idea of a worn, split configuration keyboard is not new, few prototypes have been built and tested. Due to the lack of popularity of wearable displays up to now, perhaps demand for wearable text entry devices has not been high enough to warrant research. However, in order to help determine whether a torso wearable keyboard and HMD system is a viable alternative to mobile phone devices for mobile computing, I conducted a pilot study to test input speed and gather user feedback on the prototype.

6.3.2 Prior Work

Twiddler, a one handed chording and multi-tap device, allows users to input text with one hand using various key combinations (Lyons et al., 2004). This small hand held device can easily be placed in one's bag or pocket. A similar device, the wireless Body coupled FingeRing, is a system of rings worn on each finger that also uses chording input. Though a WPM test was not conducted, numeric chord entry was demonstrated. Both of these devices are portable and can be used with an HMD. Various small form factor hand held keyboards are also available such as the Dell and Targus brand mini-QWERTY keyboards. They are thumb operated, but users tend to increase time spent looking at the keyboard as experience with the device increases (Clarkson et al., 2005). A comparison of specific WPM rates for each device will be presented in the results and analysis section later.

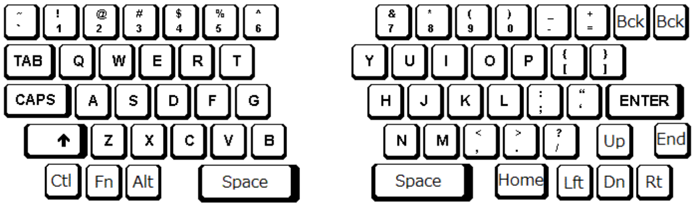


Figure 56 Torso Keyboard Layout

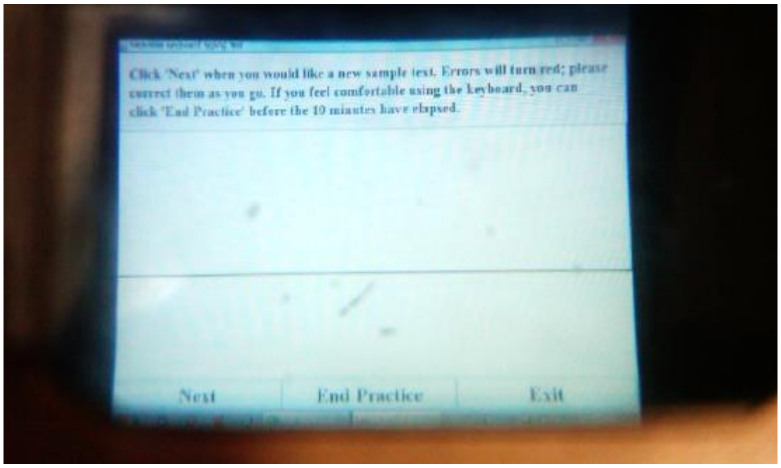
RearType is one of the latest mobile devices that incorporates a split qwerty keyboard with hands at an angled position. The keys of this device were actually located on the rear of a tablet PC, and novice user speed was compared to touch screen keyboard input speeds (Scott et al., 2010). The Half-QWERTY keyboard is a one-handed keyboard that takes advantage of two-handed typing skills. If adapted for wearable computing, this form factor could also potentially be very lightweight and one hand would remain free while typing (Matis et al., 1993). All of these devices offer a mix of benefits and different novice user speeds, however, none of these studies include testing with an HMD device. Also, since all input tests are conducted while stationary, it is difficult to determine how these devices will function in real world walking or standing situations.

My contribution includes the prototype of the torso keyboard and the analysis of its form factor for text entry. I show that it is robust, usable with an HMD while standing or walking, and for users with touch-type ability, allows for a high initial WPM speed compared to other wearable text entry devices.

6.3.3 Construction

The torso keyboard prototype was connected to a Hewlett Packard 2740p laptop computer via universal serial bus (USB), which was in turn connected to a custom HMD so that users were provided with immediate visual feedback. The wearable keyboard itself consists of several main parts, including mechanical type keys, a flexible rubber sheet, wiring, a USB keyboard controller, a thin layer of flexible plastic, and a vest as shown in Figure 55. It was important that the keyboard be somewhat flexible in case the user bends over, so keys were individually inserted into the 5 millimeter thick rubber sheet. Since user height and arm length varies, hook and loop fasteners were used for height and angle adjustment.

Layout of the keys was similar to a standard QWERTY layout but was split into two parts. The 6, T, G, and B keys, and everything left of the aforementioned keys remained on the left half of the keyboard. The right side of the keyboard included the 7, Y, H, and N keys and all other remaining keys as shown in Figure 56. Other differences from the layout of a standard QWERTY keyboard were as follows. The space key was split into right and left halves. Instead of one large backspace key, two smaller ones were included for convenience. The size of shift, enter, and space keys was reduced. Lastly, distance between keys varied based on the degree of keyboard flex.



Click 'Next' when you would like a new sample text. Errors will turn red; please correct them as you go. If you feel comfortable using the keyboard, you can click 'End Practice' before the 10 minutes have elapsed.

Figure 57 View of interface through HMD eyepiece.

The wearable monitor utilized was built from parts of an Olympus Eye-trek FMD-700 face mounted display and a Shimadzu Data Glass 2 wearable display. A single eye configuration was chosen for this experiment so that users would be able to see their immediate environment during the typing-while-walking tasks. A view through the HMD eyepiece can be seen in Figure 57.

6.3.4 Experiments

Since I sought to improve text input speed and usability for wearable computing, the experiments focused on testing those qualities. It was first necessary to develop a typing interface that could test a user's WPM through the HMD. The test interface was developed in Java, and the window visible to users was scaled to a 800x600 resolution to fit the HMD screen. The Eye-Trek HMD simulated a proportional 52 inch viewing screen at a distance of 2 meters. While typing, errors would turn red on both the original text as well as the user's input section, and would be accompanied by a short beep emitted from the laptop.

6.3.4.1 **Setup**

A total of 6 males and 1 female from 19 to 32 years of age voluntarily participated in the experiment. The only criterion for participation was that the users had to be touch type capable. Typing tasks included copying passages over 400 characters in length, consisting of upper/lower case letters and punctuation. Including spaces, WPM was calculated at 5 characters per word, and I collected data for a total of 19,200 characters. 8 passages were rotated between participants so that no participant copied the same passage for any individual task. A sample sentence from one of the passages reads: “Scarcely had the Abbey Bell tolled for five minutes, and already was the Church of the Capuchins thronged with Auditors.”

At the beginning of the experiment, users were instructed to put on the wearable keyboard apparatus and adjust the orientation of each half of the keyboard to the position that felt most natural to them. Next, participants were asked to put on the wearable monitor and adjust it until they could read text in the typing window. Participants had 10 minutes to practice typing a set of 4 passages. They were required to correct errors as they typed. If they felt comfortable using the keyboard before the 10 minutes had elapsed, they were allowed to continue to the first task. Since no mouse input was included, when the participant wished to proceed to the next task, the instructor advanced the task for them.

For the first two typing tasks, participants were allowed to sit in a straight or reclined position based on their preference. Once complete, the participant was asked to stand up and continue with the next two tasks. Several users chose to slightly re-adjust the position of each half of the keyboard at this point. Next, the user was asked to begin walking on a treadmill, and the pace was set to 1.8 kilometers per hour. When finished with the last two tasks, participants were asked to return to a sitting position. The last two typing tasks were conducted on the laptop, which has a full QWERTY keyboard. All participants finished the experiment in less than 45 minutes and were shown their own results upon completion.

To test robustness, challenges that an everyday user might experience were included in the experiment as follows:

- Although the typing tasks were conducted in English, most participants were non-native English speakers. Participants' native languages included German, Greek, Japanese, Vietnamese, Chinese, and English.

- Instead of using a high resolution computer monitor or projector from which to read text, participants used an HMD configured for left eye viewing.

- All tests were conducted in a public seating area so that background noise such as occasional conversations and footsteps were audible.

6.3.4.2 **Results**

Across all tasks with the wearable keyboard and HMD, average input speed was 30.14 WPM. The average laptop input speed was 56 WPM, which means that users were able to achieve 53.8% of their baseline. A repeated measures Analysis of Variance for all conditions shows a significant effect between the wearable and laptop keyboards ($F_{(1,6)} = 20.56$, $P<.05$). There was also an effect between first and second paragraphs ($F_{(1,6)} = 12.62$, $P<.05$), so I can predict an increase of input speeds with practice. A breakdown of average WPM speeds by task (sitting, standing, walking) and by trial (first or second run within a task) for the wearable keyboard is shown in Table 8.

Table 8 Results according to task and trial (in WPM).

Task	1st Paragraph	2nd Paragraph
Sitting	30	32.8
Standing	27.7	35.5
Walking	25.2	29.4

6.3.5 Discussion

Despite various challenges related to language and vision, the fact that participants were still able to average over 30 WPM in various conditions demonstrates the robustness of the torso wearable keyboard. For similar devices where novice user data was recorded, a comparison of typing speeds is shown in Table 9. These results do not include the use of predictive typing or lexicon based software. I consider a novice user's average speed of 30.1 WPM to be a significant improvement over most other devices for use with touch type capable individuals, especially considering it was tested with an HMD and that this average includes standing and walking tasks. Although touch typists could already be considered experts or as having prior experience with my keyboard, this still means that it could be used for HMD input with a large existing user base and minimal training. Those who can’t touch type would have to learn, but considering the popularity of QWERTY touch typing for office work, my device is well suited for a large number of office workers who must touch type on a day to day basis. Participants also stated that a number of conditions may have decreased their performance during the test as follows:

- Trouble reading text due to the limited size and clarity of the HMD. For example, several users were sometimes unable to distinguish between lowercase L and I.
- Unfamiliarity with words due to lack of English ability.
- Several users were used to typing the B key with their right hand, though it was on the left half of the keyboard.
- Backspace key was difficult to reach.
- Difficult to adjust keyboard height to appropriate level.

Other comments not necessarily linked with typing speed were that the prototype was too heavy and bulky, and that a belt or pant pocket attachment would be beneficial. Based on observations, keys typed with the pinky and pointer finger were more difficult to type on, suggesting that rotation of the arms and wrists affected performance (Serina et al., 1999). One of the main drawbacks of the torso keyboard is that users with no touch type ability will need time to learn how to type without looking. It is impractical to constantly look down at one's torso while standing or walking. In contrast, the wearable form factor prevents the user from having to constantly store the device in a pocket or bag.

For those that can touch type, the ability to enter text quickly while walking will increase mobility, especially since the user can look straight ahead while entering text. Though social acceptance of the torso keyboard has yet to be tested in an office or commuting environment, participants seemed to have fun using the device and several people asked when it would be available in stores. As with any new technology, it is likely that the torso keyboard will meet with initial resistance, followed by gradual acceptance over time. Based on user feedback, I have already developed a wireless implementation of the prototype as shown in Figure 58. It is thinner and weighs under 300 grams, much lighter compared to 2.1 kilograms for the first prototype. Though it still uses thin ribbon cables to connect to the wireless controller, wireless modules could be included on each side of the keyboard in a final product. With this second prototype, I would like to observe how obstacles in a user's path and faster walking speed affect WPM and safety.

Table 9 Typing speed results for novice users of various input devices.

Device	**WPM**	**Sit, Stand or Walk**	**Type**
Twiddler[5]	8.2	Sitting	Hand Held
Phone keypad[7]	9.1	Sitting	Hand Held
Stick keyboard[4]	10.4	Sitting	Wearable
RearType[8]	15	Sit, Stand	Hand Held
Torso keyboard	30.1	All 3	Wearable
Mini-QWERTY[1]	34.3	Sitting	Hand Held

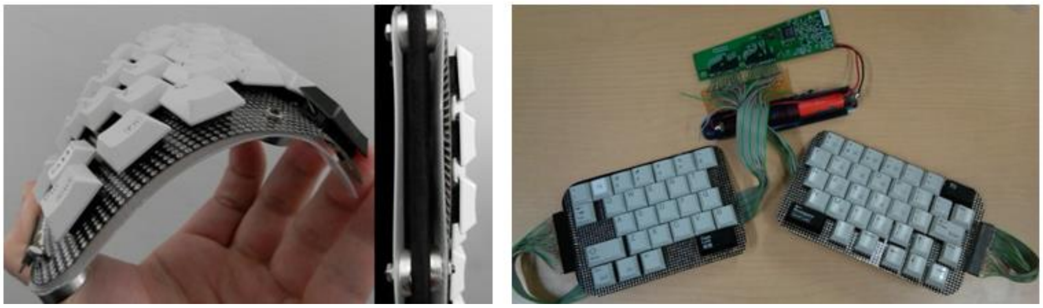


Figure 58 Second keyboard prototype showing the flexed board and the split, wireless configuration.

6.4 Augmented Reality Memory Assistance

In our everyday lives, bits of important information are lost due to the fact that our brain fails to convert a large portion of short term memory into long term memory. As a step towards building better memory assistive technology through AR, I propose a framework that uses an eye-tracking interface to store pieces of forgotten information and present them back to the user later with an integrated HMD. This process occurs in three main steps, including context recognition, data storage, and AR display. The system's ability to recall information is demonstrated with the example of a lost book page by detecting when the user reads the book again and intelligently presenting the last read position back to the user. Two short user evaluations show that the system can recall book pages within 40 milliseconds, and that the position where a user left off can be calculated with approximately 0.5 centimeter accuracy.

6.4.1 Introduction

It has been long known that humans often fail to convert short term memory into long term memory, and are inherently forgetful. We often mistakenly judge certain events as being unimportant, but which turn out to be important at a later time or in a different context. To help cope with this memory deficiency, technology has been used as a form of cognitive offloading to assist and sometimes even function as a substitute for memory intensive tasks. Good examples include digital calendars, reminder systems, life logging applications, and the use of search engines for information not committed to long term memory (Belimpasakis et al., 2009, Hodges et al., 2006, Maus et al., 2013). My research builds on this idea by augmenting memory through the use of eye-tracking and an AR display. When a user returns to the situation in which a memory occurred, eye gaze can be used to detect context and more accurately present the user with previously stored information. Eye tracking is first used to identify a user's point of attention and to outline an area for recognition, such as text or an environmental object. That text or object is then inserted into a database along with relevant tags such as date, time and location. "Memories," represented by an array of contextual and temporal tags in the database, can be recalled later with keyword searches or object detection triggers.

Examples of applications of this technology include recalling items such as forgotten page numbers in documents, the location of misplaced keys, or patient information prior to surgery. Interfaces such as this one also have the potential not only for consumer use, but for use with clinical patients suffering from memory related illnesses such as vascular dementia or Alzheimer's disease. Below, I describe a general framework that facilitates the encoding of temporal events into digital form, a system that can help a user recall a lost book page and a specific implementation that can encode an event, such as placing one's keys on a desk, into the database for later recall. Though a variety of implementations within this framework are possible, I chose page recollection and simple event storage since they are prime examples of how this framework can translate to practical application.

6.4.2 Prior Work

One widely explored field of research related to memory is that of physical systems that serve as memory aids. One such example is the SenseCam, which takes intermittent photos throughout the day and serves as a retrospective memory aid (Hodges et al., 2006). Detailed studies using the SenseCam show that memory can be improved by reviewing images taken by the system, especially long term memory (Sellen et al., 2007). A similar device called the EyeTap has been used as a form of capturing life experiences and sharing these experiences with others (Mann et al., 2005). Although a large number of other software memory aids such as calendars and reminder systems are available, a majority of them only exist as mobile or smartphone based applications. Several systems are also available that utilize sensor data in order to extract context. One such system by Belimpasakis proposes a client-server platform that enables not only life logging, but richer social experiences by extracting more meaningful contextual information from data (Belimpasakis et al., 2009). The above systems all have the potential to be combined with or improved by various models for memory and decision making, such as those proposed by Hutter (Hutter, 2005). They also fall into the broader goal of creating a complete database of all life events (Belongie et al., 2002).

Another set of closely related studies are those which use computer vision and eye tracking for context recognition. One major branch is the study of object recognition, which can be conducted using hysteresis, feature tracking, and other algorithms (Lowe et al., 1999, Toyama et al., 2012). This type of method can help with context recognition since it has the potential to extract semantic information from objects in one’s environment. In addition to recognizing objects, location can also be extracted using gaze and other sensors (Sonntag et al., 2013). In conjunction with HMD systems, activity can also be recognized using other types of mobile sensors (Ravi et al., 2005). Once context, location, or other relevant content has been determined, information visualization methods can be used to place the information in a relevant location in the environment (Orlosky et al., 2013). This can prevent information from becoming a distraction, and can make recalled information easier to view. My framework uses a combination of elements from life logging, context recognition, memory models, and AR in order to assist users with event recall.

6.4.3 System Framework

A 3D gaze tracking system combined with an HMD that does not require the use of external tracking or projection hardware is used. The device is composed of a pair of eye tracking goggles, custom 3D printed attachment, and HMD. The devices are all connected, and can be calibrated as a single system.

6.4.3.1 **Hardware**

To start, I needed an apparatus for eye and vergence tracking that could be used simultaneously with a head mounted display placed near the user’s eye. A pair of SMI Eye

Tracking Goggles was selected, which can be worn like glasses and leave enough room to attach an HMD. The HMD part of the system consists of an 800 by 600 pixel AirScouter HMD, which includes digital input via USB and depth control. The focal depth can be set from 30 centimeters (cm) to 10 meters (m).

In order for gaze to be measured appropriately in the HMD, a user's eye convergence must be consistent and eye tracking hardware must provide enough accuracy to ensure consistent gaze on a target object of interest. In addition, it was necessary that the distance between the tracker and HMD remained the same during use.

In order to provide information back to the user in an intelligent fashion, in many cases digital text needed to be aligned with objects in the scene. In the case of recalling a book page or sentence in a document, text and pointers must be displayed in line with the targeted object and text. First, when a scene image is taken from the camera, the image is blurred by a Gaussian kernel and thresholded into a binary image in order to detect the centroid of each word region. The retrieval process is done by matching extracted features to the features of books, documents, and other media previously stored in the database (Toyama, 2013). Since an image based method is applied, a variety of different paper mediums, fonts, and sizes can be dealt with. By matching the features between the scene image and the retrieved database image, the homography between them is also calculated. Based on this homography, the pose, rotation, and transformation of text in the scene image can be estimated. This data can also be used both for HMD calibration and correct projection of overlaid data.

6.4.3.2 **Software Flow**

Data processing within this framework primarily occurs in one of three steps, as described below. The first phase is interaction, where the primary sources of input are the position extracted from the eye-tracking interface, the environmental image from outward facing camera, and sensors such as GPS, accelerometer (for determining activity through methods such as those by Ravi et al.), and system time (Ravi, 2005). The second step is the encoding of this information into the database. Input data is stored in the database as an array of searchable keywords, and elements like time are stored as a chronological array. Finally, recollection of events is triggered by user initiated keyword search or by recognition of current context, and relevant database entries are displayed back to the user through the mixed-reality display.

6.4.3.3 **Database Design and Storing Events**

Once an object or set of objects has been recognized, it must be stored in the database in a particular context. One important dimension for context is time, since events that occur closer to one another are likely more closely related. Time is also important in human memory, like our ability to remember procedural tasks or sequential events better than randomly distributed ones. This is the reason why many people must sing a song from the beginning in order to recall a particular phrase in the song. Other dimensions include semantic relevance, physical location, and custom input for more specific applications. These dimensions can also be cross-referenced to improve recall. Though the number of dimensions could be expanded with additional implementations, the current database elements include 1) an event, which represents the essence of the memory, 2) time, which is the moment in time when the event occurred, 3) location where the memory occurred, 4) semantic context, which includes any available information extracted from OCR or other contextual data extracted from sensors, 5) keyword, which represents an optional additional relevant contextual cue, and 6) an arbitrary field for use with specific implementations, such as the book page recollection algorithm. Though the mechanisms behind human memory are not fully replicated in the framework, these methods can serve as rough metaphor for basic storage and recall of past information.

Database queries can be manually engaged by a user, or automated based on triggers from a certain event or idea. If a user were searching for his or her keys for example, he or she would input “keys” as a keyword search and would be presented with a list of terms from the database contextually related to the word “keys,” such as room numbers or objects detected in the immediate vicinity or time frame of the nearest occurrence of keys. This method is comparable to personal information models such as those proposed by Maus et al., but takes advantage of augmented reality for reduced interaction and faster presentation (Maus et al., 2013).

6.4.3.4 **Information Presentation and View Management**

Once information has been recalled from the database, it must be presented to the user in an intelligent way so that it is in context and does not induce confusion. In the example of finding the last sentence a user was reading in a certain book, simply displaying the first word of the sentence in the HMD would not be enough for a user to find his or her place quickly. Instead, the system uses a document image retrieval and projective calculation to determine the position of the book, and appropriately displays a notification or pointer to where the user left off in the real world. Finally, view management can move resulting notifications to ensure that recalled information does not interfere with reading, walking, searching, or other visual tasks (Orlosky et al., 2013).

Here I present a software implementation of the framework which accounts for a certain type of cognitive task. Like many easily forgotten events, leaving a reading task without marking the page is a frequent occurrence. By implementing one type of recollection method within the framework, I can solve this problem. Using the same steps outlined in the framework section, this particular method detects when a user is reading a specific document or book, searches the database for any memories related to reading that particular book, and displays navigation cues to the reader to show the page and location where he or she last left off, as shown in Figure 59. In addition to displaying the correct page, pointers show the user the direction of their last reading position, and a line is displayed under the last word read.

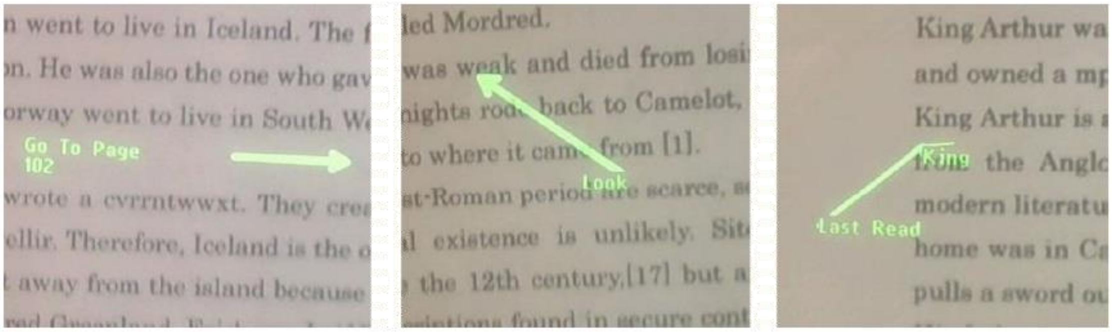


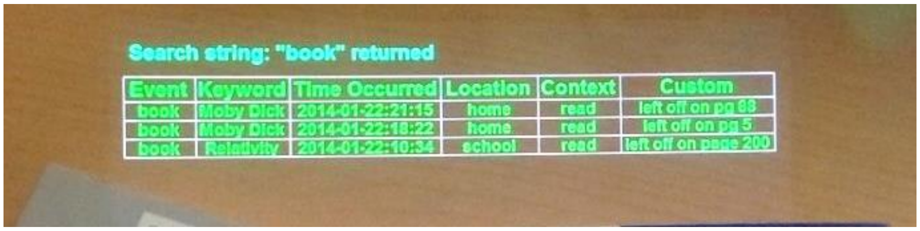
Figure 59 Images showing recognition of an incorrect page (left), correct page with bookmarked text located above the HMD viewing field (center), and correct position with pointer (right).

With respect to memory logging and recall mechanisms, to provide a visual representation of how entries are recalled from the database, Table 10 shows how entries from a single day would appear in the database, and Figure 60 shows a view through the HMD showing a corresponding list of results using the keyword "book" as a search string queried from that database.

The keyword search would be narrowed down further upon adding additional search strings such as time range or location as shown in Figure 60. In the case of visual book recognition, instead of presenting search results, the document recall algorithm takes over, and displays the page and navigation instructions from Figure 59. To provide a simple evaluation of book page recall, two short experiments were carried out. The first was designed to measure the time it takes to recognize a page when a user first looks at a document, and the second was designed to determine how accurately the exact reading position could be measured for re-display.

Table 10 Sample database entries for a single day.

Event	Keyword	Time	Location	Context	Custom
book	Moby Dick	2014-01-22:21:15	home	read	pg88&x36&y773
book	Moby Dick	2014-01-22:18:22	home	read	pg5&x120&y150
magazine	Modern Art	2014-01-22:01:34	library	view	null
book	Relativity	2014-01-22:10:34	school	read	pg200&x52&y318
memo	groceries	2014-01-22:07:21	home	view	null



Search string: "book" returned

Event	Keyword	Time Occurred	Location	Context	Custom
book	Moby Dick	2014-01-22:21:15	home	read	left off on pg 88
book	Moby Dick	2014-01-22:18:22	home	read	left off on pg 5
book	Relativity	2014-01-22:10:34	school	read	left off on page 200

Figure 60 Segment of an image taken through the HMD viewing screen of returned search output.

Time-to-recall: The first experiment was conducted by asking 10 users to wear the display system. Each user was then presented with both a document presented on a computer monitor and a printed sheet, both of which had the same size and text. They were then asked to read each document as if they typically would any other type of text, and the recall algorithm was applied to each frame throughout both reading tasks. Reading angle was also measured for each participant to test whether I could still recall the text despite different viewpoints. Results show that for both the digital and physical documents, the recognition accuracy for each frame was 100% for reading angles between 70 degrees and 90 degrees. There was only a 0.54% decrease in accuracy for viewing angles between 50 degrees and 70 degrees. Other informal experiments showed that for over 50 degrees of deviation from vertical, accuracy of recall decreases rapidly. A reading position test was also carried out, where another set of 13 users were asked to read through a document and pause at four different words over the course of two minutes. For each word, the distance between the center of the requested word and the point provided by the eye tracker were measured. On average, the deviation from each word was approximately 0.5 cm across all participants for the two minute period, and showed a minor decrease in accuracy over the first minute. This distance is equivalent to either one line of text in the vertical direction or one to two words in the horizontal direction, meaning that a user would never have to read more than one or two lines of text away from his or her last reading position. With this level of accuracy, we can conclude that recall of page and position is effective for general use.

6.4.4 Discussion

In addition to the general recall of information, I have also explored the possibilities of the eye tracker and HMD setup for recalling patient faces and virtual display of patient records (Sonntag, 2015). A generalization of this approach is the exploitation of eye movements in the context of more complex activities for which the role of vision has yet to be explored. New application domains should take daily activities into account and provide for cognitive assistance in those activities. The aim of these current studies is to determine the potential impact of such cognitive assistance for specific user groups in both medical and consumer applications. The setup can also potentially be used to interpret the center of gaze and fixations of dementia patients, which can be used to recall assistive information from the database.

6.5 Cognitive Monitoring and Evaluation

In addition to memory assistance, it is also very important to be able to study and evaluate an individual's cognition in a controlled manner. Recently, simulations for monitoring, evaluation, training, and education have started to emerge for the consumer market due to the availability and affordability of immersive display technology. In this work, I introduce a virtual reality environment that provides an immersive traffic simulation designed to observe behavior and monitor relevant skills and abilities of pedestrians who may be at risk, such as elderly persons with cognitive impairments. The system provides basic reactive functionality, such as display of navigation instructions and notifications of dangerous obstacles during navigation tasks. Methods for interaction using hand and arm gestures are also implemented to allow users explore the environment in a more natural manner.

6.5.1 Introduction

In recent years, intelligent technologies have been proposed as a tool for assisting with education and training for a wide range of fields (Chahudhri et al., 2013, Petersen et al., 2012). A number of these assistive technologies are designed for tracking and remedying of cognitive disabilities, which often manifest in different ways. For example, a dementia patient may exhibit wandering behavior, the exact nature of which is difficult to determine since his or her exact movements and field of view may not be recorded. To address these challenges, I introduce a prototype simulator that can provide better mechanisms with which to study various types of activities and cognitive states in the domain of pedestrian safety. In order to improve monitoring, analysis, and training for such individuals, I combine a number of interactive technologies such as the Oculus Rift, LEAP Motion, and the Myo by Thalmic Labs. These tools enable hand tracking and sensing of arm motion and muscle activity that can be used for interactions such as button presses and walking within the simulated environment. The virtual environment is created with Unity 3D, and the simulation itself is viewed through the Rift to provide an immersive experience.

Additionally, two types of hand and gesture devices are used for both input and monitoring when users are engaged in virtual tasks. These devices allow users to directly interact with the simulated environment in a more natural way, and can also be used for online monitoring and post-analysis of the user's activities and movements relative to his or her field of view. Overall, the system functions as an adaptive test bed for studying impairments and at the same time facilitates better interaction. This work describes the prototype system, where users can complete tasks in the virtual environment in different conditions, use hand tracking and integrated muscle sensors for movement, and interact with the safety notifications generated by the system. Tasks include navigating through the virtual environment in different conditions and reaching a specified location. Users have to deal with both traffic and obstacles to complete each task successfully.

6.5.2 Prior Work

The recent development of 3D tools and technologies for gestures, interaction, and control has led to new possibilities for 3D interactive applications and immersive environments (Plemmons et al., 2014). One such example is hand-based interaction, where a user can directly control 3D objects with his or her hands in both virtual and augmented reality (He et al., 2014). Haptic devices can also be used to interact and provide feedback for the user, which have been employed in simulations and for control of robotic systems (Bar-Cohen et al., 2003). Head worn displays have proven to be instrumental for implementing immersive simulations of situations with present danger or limited visibility, which provides further motivation for us to use virtual content to help evaluate perception of a dynamic environment (Orlosky et al., 2014). Even more recently, interacting in a virtual immersive display has proven to be useful in treating amblyopia, as shown in (Blaha et al., 2014).

These types of interactions and simulations have also had more specific applications in education and medicine. In 2010, a simulator was developed that showed benefits for training the elderly with respect to pedestrian safety. Gu et al. have more recently proposed the combination of semi-immersive virtual reality (VR) with intelligent tutoring approaches in order to support children learning pedestrian safety (Gu et al., 2014). Research shows that children, the elderly, and the intoxicated are the most endangered categories of pedestrians; therefore, technology to evaluate and support these groups is in high demand (Whelan et al., 2008).

6.5.3 Cognitive Monitoring Framework

6.5.3.1 **Simulation Hardware and Software**

With this system, I seek to improve on prior simulation technology to provide a more interactive and immersive environment. Moreover, I can extract more information about user actions than in more simplistic setups, which can be utilized for more thorough analysis and study of an individual’s behavior.

As an immersive display, I use the Oculus Rift DK2 head worn display (HWD), which provides stereoscopic images to the user, conducts six-degree-of-freedom head tracking, and gives access to video and position data streams so that I can monitor the user’s field of view and head orientation. Less immersive setups, for example those based on single or multiple monitor displays, may fail to provide a real sense of danger and/or spatial understanding during simulation. CAVE systems, which project images onto walls that surround the user, are also immersive, but are not as affordable and maybe difficult to obtain for a majority of potentially interested stakeholders such as educational institutions and healthcare facilities.

The simulation itself is implemented in Unity 3D, which enables generation of high resolution virtual worlds and an improved sense of presence. The simulation is set in a manually constructed urban environment, as shown on the left of Figure 61.

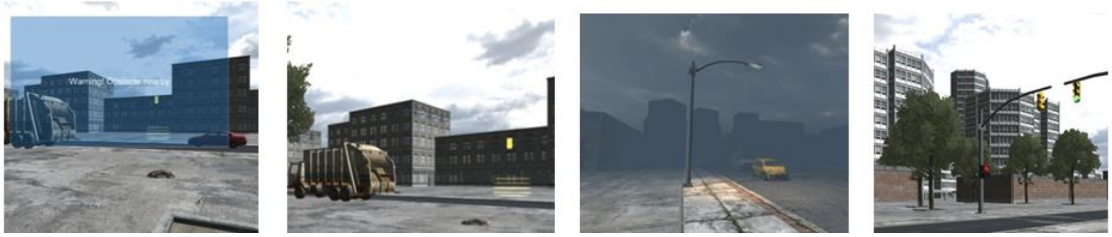


Figure 61 Simulated environment showing the reactive warning system alerting the user to a tripping hazard (left), obstacles and waypoint (mid-left), varied visibility to simulate a foggy or dark night (right-mid), and an interactive traffic light (right).

The environment contains many stationary objects, such as trees, benches, rocks, roads, curbs, and other obstacles (e.g. parked cars and road curves), that might be of concern for an elderly patient or an individual with motor impairments. Additionally, various dynamic and interactive objects such as cars, traffic lights, waypoints, and buttons are implemented so that the user can complete tasks while dealing with realistic environmental obstacles. As a part of the simulation, reactive alarms are installed on potentially hazardous objects. For example, when a user fails to look at an object near his or her feet, an alarm appears in the display to alert him or her of the object, as shown on the left of Figure 61. The alarms are activated only when a user fails to notice an object. For example, an individual may not look down at a curb or rock in his or her path. By checking the distance to the object and current camera frustum, I know whether or not that object has entered the user's central field of vision. If the person has come too close to the object without it entering his or her field of view for some time, the alarm is triggered in the display. Though this alarm system is mainly designed to provide feedback on a user's cognitive ability and learning, it can also be used to study reactions to augmented elements in the real world. The simulated alarm also resembles a notification that someone might see in an optical see-through HWD used outside, so I can also gather feedback about virtual text notifications that might be presented in a mixed reality situation.

More importantly, by recording the user's head movements, I can measure reaction time, whether or not certain objects are noticed, and general movement tendencies. The change in response over time (i.e. learning) to certain alarms or notifications can also be used to monitor the patient's mental state, or gather data about any cognitive impairments that may be present. Based on the age, conditions, and ability of the user, tasks can be modified to require a higher level of cognition, such as adding a blind curve, an obstacle, haze, or darkness as shown on the right of Figure 61. Users must then think ahead in order to predict whether the situation poses a greater danger than when all traffic can be seen clearly from a distance. Consequently, the ability of the user to predict or think at a higher level can be analyzed since I have access to field of view data within the virtual environment.

6.5.3.2 **Physical Interactions**

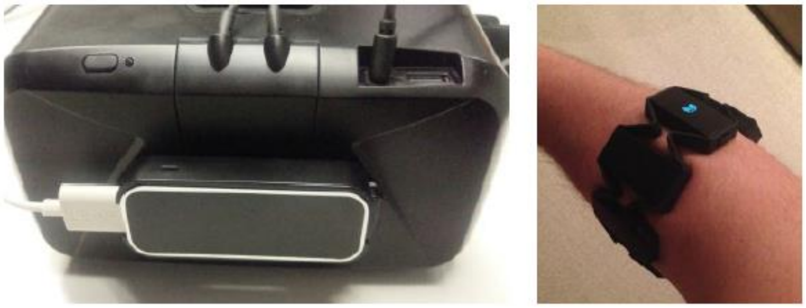


Figure 62 Oculus Rift with attached LEAP Motion for hand interactions such as button presses (left) and the arm-worn Myo to facilitate walking via natural arm movements (right).

I also needed a way for the user to physically interact with the environment. First, in order to move through the simulated world, he or she must have a method to engage walking. Many other implementations utilize keyboards or controllers for movement (Gu, 2014), but I sought to have something more natural, especially since the target users are individuals who may have a cognitive impairment. To accomplish this, the Myo was selected, which is an arm-worn band by Thalmic Labs that contains sensors to measure movement and muscle activity. Since the device is arm-worn, as shown in Figure 62, it can be used in a non-invasive manner, and can take advantage of the user's natural arm movements as a form of input. In particular, I utilize the natural swinging of the arm to facilitate movement. Since a person's arms swing while walking, this becomes a much more familiar way of interacting than by pressing keys on a keyboard. Faster arm swinging corresponds to faster movement within the environment. Although walking machines for virtual reality such as the Omni Treadmill are available, they are still expensive and not portable. For the purposes of research and studying patients at a healthcare or research facility, the Myo provides us with a small, inexpensive input device.

Second, users must be able to physically interact with virtual elements in the simulated world, for example, by pressing one of the signal crossing buttons located on the traffic lights shown on the right of Figure 61, or signaling a taxi. For this purpose, I use the LEAP Motion, which is mounted to the HWD and provides near-field hand tracking and is ideal for on-demand interactions in our environment, specifically, button presses for road crossings. The role of these procedural tasks is especially important for evaluating user behavior and cognition. The user also sees his or her hands inside the simulation; thus, a more realistic physical interaction is achieved. Although haptic feedback via Myo has not been yet implemented, I plan to incorporate a tactile vibration upon a button press as a part of future work. The Leap-motion Attached Oculus Rift and Myo are shown in .

The purpose of using these tools is two-fold. While I need to provide the user with natural interaction, I also want to record as much data as possible about movements or judgments that may signify an impairment or general lack of cognition at a certain time. Examples include forgetting to press a button altogether, excessive search behavior, or severely delayed reactions to an important stimulus.

6.5.4 Discussion and Summary

Arm, hand, and finger gesture recognition are of particular interest for future natural user input scenarios with implicit gestures and, as a byproduct, for reducing cognitive load (compared to explicit gesture, which a user has to learn). Implicit gestures need to be grounded so that the human computation part (recognition and understanding of human gestures) can be achieved. Data collected by this kind of simulator has the potential to produce better recognition accuracy and hence more realistic VR scenarios and user behavior capture and interpretation. Collected samples of usage data, such as gestures or actions used to complete tasks in the simulator, can be labeled post-task, and categorized via classification algorithms. Next steps include the creation of a 6D motion gesture database for implicit gestures and the application of new spatio-temporal event classification algorithms (Jeni et al., 2014).

Reliable implicit gestures could also be used for gesture-based disambiguation of user intents. In addition to the already implemented alarms, providing reminders about consecutive stages in individual activities such as "press the traffic light button" to compensate for a lack of situational awareness can be useful for more severe dementia patients, and for caregivers who are evaluating the patient. In addition to evaluating patients, the virtual reality environment may develop into a safe, cost-effective, and engaging approach for future immersive training environments of dementia patients where training implicit gestures could help improve the performance in daily life (e.g., pressing buttons on home appliances to remember their functionality). This training could be greatly beneficial for increasing a user’s retention of situations and the implicit usage of gestures for controlling electric and mechanical machines (Gupta et al., 2008). In addition to traffic scenarios, the reminders or alarms could also be evaluated for scenarios that may be difficult to evaluate in the real world due to privacy issues (e.g. in a bathroom).

6.6 Summary

In summary, this chapter outlines a number of more specific applications to enhance usability and safety in various situations. I first presented a magnetic tracking system and OCR based localization method. Several problems with using magnetic tracking in various environments including shifting and speed variation are revealed, and I provide a solution to shifts using normalization. I then outline appropriate usage of the fused system and show through two pilot experiments that a fused system is feasible for emergency scenarios.

Next, I built and tested a prototype of a clothing-wearable text input device called the Torso Keyboard. In my experiments, users were able to achieve relatively high WPM rates on various typing tasks while walking and standing. Given these results, I conclude that when used with an HMD system, the torso wearable keyboard is a potential alternative to cell phone devices for mobile text entry tasks.

Then, I propose the use of a combined eye-tracking HMD interface for detecting context, storing events into a database, and virtually presenting those events back to the user at a later time. Within this framework, I implement both the database for storing and recalling events, and a more specific method for recognizing documents, which virtually projects a pointer to the last location in the real world where the user left off. I then conduct two short evaluations testing the accuracy of document recall and reading position, finding that both are effective for practical use. This system can function as a cornerstone for the development of other context sensing AR interfaces, and I hope it will encourage further research on memory assistive technology.

Finally, I introduce a simulator designed to improve monitoring interaction for and analysis of cognitive abilities in a virtual environment. I construct a replica of a suburban environment, and provide a number of navigation tasks within the environment. This environment is viewed through the Oculus Rift, and includes integration of the Myo and LEAP Motion sensors into the framework. This allows for more natural input, and enables collection of more detailed data and feedback for individuals such as dementia patients. I hope this environment will promote more detailed study of cognitive abilities and can be used in other contexts, such as education and training.

CHAPTER 7

Summary of Findings and Discussion

7.1 View Management

Chapter 1 of this thesis outlines the division of virtual information into two primary types, user centric and environment centric. In view management, these new classifications will be essential to the proper management of content as AR becomes more commonplace in the real world. Unlike traditional view management strategies, which focus on environmentally registered objects such as markers, labels, and static augmentations, user centric content must travel with the user as outlined in Chapter 3. Because of this mobility and constant dynamic change, view management methods that assume a label is affixed to a model, rather than to the user's display, are not adequate to handle the management of user centric text. Experiments testing distraction, fatigue, and virtual simulation sickness also deserve further testing.

While the methods in Chapter 3 provide several fundamental strategies for dealing with mobile user centric text, it does not cover all applications and situations. This leaves the field open for other researchers to develop new methods for more specific applications that may require different management strategies and different displays. As mentioned in the introduction, the methods presented here are primarily designed for HMD based devices, but can be extended to hand held displays. Moreover, there is still a significant opportunity to integrate interactive techniques with adaptive management to form an even more cohesive experience for the user.

In addition, the case still remains where both user centric and environment centric information are merged into the same FOV. Aside from outliers where the lines between user and environment centricity are blurred, the case that will start to occur more and more frequently is when the two information types are displayed simultaneously. For example, a notification and newspaper article (user centric) may be present on the display screen, and at the same time building labels and ratings may be displayed for restaurants along the user's path. This mixing of the two types of information with the view of the real world adds additional complexity to the view management required for that display.

Finally, I highly recommend that other researchers focus on the mental process by which algorithms are generated when trying to solve a problem. For example, when coming up with the Dynamic Text Management algorithm, I first thought very critically and introspectively about how humans would manage content if placement choices were translated directly to reality. When asking myself where I would place text, I found that my answer was always on a wall or uniform surface. After conducting a literature review about characteristics that resulted in better reading performance (such as contrast manipulation), it became more and more evident that humans tended to fit content to a location in the environment that was both uniform and dark, therefore providing the highest contrast and viewability. Although human preference is determined by a plethora of various logic and experience, the Dynamic Text Management algorithm shows that when it comes to content placement, many of these preference tendencies can be abstracted into algorithmic form. The experiments comparing human and algorithmic placement tendencies provide significant evidence for this point.

Still, each human has his or her own personal preference, as was revealed by the experiments testing the Halo Content algorithm. Some people simply do not want any virtual content at all when engaged in a conversation with another individual. While an algorithm can generally reduce invasiveness or improve visibility, customization and parameter tuning are just as important. Some of this preference to remove all text from the scene can be attributed to bias, since a number of people are taught not to use technology when engaged in interpersonal interaction. For example, it would be considered rude to pull out a phone during an interview or business meeting and take your attention away from the speaker. This again emphasizes the importance of customization or personal parameters for managing content. Users who are skilled at multitasking or do not find virtual content distracting can benefit greatly from a view management algorithm that keeps content visible, but out of the direct line of site. Others may need an algorithm that can remove text completely to allow for full concentration.

One such situation where a completely clear view of the environment is beneficial is navigation. To ensure a clear view of the environment, I have found that eye-tracking is one potential solution.

7.2 Integration of Eye Tracking

When virtual content is present, automated methods for placement can handle some problems associated with viewability, but not all. Using eye tracking to determine the user’s attention is one method by which we can manage some of these other situations. For example, a view management may be used to avoid cars are bikers in the environment using a movement scheme. However, if the user is walking down a sidewalk where there is little chance of crossing the path of a car, the content management algorithm may unnecessarily move content. If eye tracking is also integrated into this process, it is possible to determine when to show or move content based on whether the user is actually looking at the car or not.

As discussed in the attentive interfaces section, automatic dimming or content removal can be done very quickly with good eye tracking hardware. Additionally, since any deviation from the current focal plane for a short period of time can be considered a change in focus, using depth tracking is a fast method for interaction, and can differentiate between objects in the same line of sight, but at different depths. One other important finding in this thesis is that despite only having a single monocular depth cue, both eyes still verge. This raises a very interesting question: To what extent would other monocular depth cues cause vergence of the eyes? It may even be possible to use experiments testing resulting vergence resulting from monocular depth cues to test the strength or importance of those cues.

Furthermore, additional experiments testing the use of icons in a user's peripheral vision would be very beneficial, assuming wide FOV optical see through technology has become widely available. This will allow testing of distraction and other methods for interaction in more detail. Additionally, thorough testing of the interaction methods in outdoor AR environments is necessary to determine usability. I also intend to utilize the outward facing camera on the eye-tracking interface to determine appropriate locations for icons in a user's field of view or in the environment. Recordings of user eye movements coupled with the outward facing camera will likely provide further insights into the behavior of the eye.

7.3 View Manipulation

Although vision enhancement can be user centric much like virtual notifications or messages, augmenting vision is better to leave in its own category. A vision enhancement may also be very closely tied to the environment, like a zoom window for example, but resulting information presented to the user is still very dynamic. It is more beneficial to think of this kind of information as a transformation of environmental information into a user centric space. As such, different methods are required for engaging and managing enhancements. From the experiments conducted with the Fisheye Vision and ModulAR displays, we can conclude that visualizing an enhancement should be determined based on application. Moreover, the method by which augmentations are engaged must be flexible. A firefighter that needs to use an on demand navigation system will have very different needs than a botanist who is observing plant growth from a distance.

The Fisheye Vision display has also opened up a new area of study with respect to stereoscopy. The fact that two radially distorted images are perceived as a single binocular image by the human brain is very interesting. This fact raises questions as to how far we can distort (or compress) an image into the human field of view. It will also be very important to determine the effects of such compression on esoteric motion perception in mobile environments. It is likely that such experiments will need to be tested in completely virtual or highly controlled environments first to study user safety.

7.4 Integration

Though chapters 3, 4, and 5 present a variety of strategies for managing content and field of view in different contexts, many of these methods can potentially be combined for an even more effective interface. For example, the Dynamic Text Management presented in Chapter 3 would still be effective if used with either of the displays in chapters 4 or 5. Even if text is moved off to a different, visible location in the background, dimming the text when the user is concentrating on the environment can still improve awareness and visibility of potential obstacles. In the case of the ModulAR display, zoomed content such as the Snapshot view presented in 1.1.1.1 could also be managed using the same strategy of finding dark, uniform regions in the background on which to display the magnified window.

Though the ModulAR display cannot yet replicate focal depth, the strategy behind the focus based interface proposed in Chapter 4 could work with the ModulAR display based on convergence. Alternatively, if a multi-focal wide FOV optical see-through or video see-through display becomes available, the dimming strategy will become even more effective due to the increased angular range of vision. In other words, wearers of the display will not be confined to a small window in which convergence and accommodation can be measured. Though the software described in this thesis has been closely integrated into each piece of hardware, the strategies and algorithms used could in fact be decoupled from the hardware and applied to other head mounted devices regardless of whether they are optical or video see-through or monocular or binocular if desired.

CHAPTER 8

Conclusion

In this thesis, I propose a variety of solutions and strategies to improve the safety and usability of augmented reality. More specifically, I propose the use of adaptive view management techniques, improved attentive interfaces, and manipulation of a user's view space.

First, I propose Dynamic Text Management, which improves upon previous methods for displaying text in AR, combines those methods into a single framework, and becomes an adaptive system for managing text that can run in real time. The system allows text viewed through an HMD to move with the user as the user travels throughout various environments and maximizes times where text is viewable by finding dark, uniform regions. To deal with interpersonal interactions, I introduce Halo Content, and find that it can reduce the invasiveness of virtual augmentations, while still providing easy access to content for the user. Since face and body detection algorithms are not perfectly robust, I also provide a way to account for facial persistence, which results in more consistent management and smoothed movement. These frameworks can also be used with other object or feature recognition algorithms, making them scalable and applicable to augmented reality applications in a wide variety of other fields. These methods can serve as a foundation for managing user-centric text and provide useful insights for other content or view management research.

Since content management via placement does not deal with situations in which a completely clear view of the environment is necessary, I then propose a framework that can remove or dim content base on the user's eye gaze patterns. This improves upon prior works by developing a more natural interface for interaction with virtual elements. The system includes a prototype hybrid multi-focal plane eye-tracker/HMD, which facilitates methods to automatically close, remove, or dim distracting content from a user's field of view. I then conduct several experiments to test the viability and effectiveness methods for interaction and find that users can accurately focus on virtual content displayed on different focal planes despite having few monocular depth cues. More importantly, this focus can be tracked robustly for use with other similar HMD interfaces, and distracting text can be quickly removed in situations that require immediate attention.

Lastly, to deal with displays that have limited field of view, I introduce Fisheye Vision, a method for expanding a user's effective field of view in see-through displays using fisheye lenses. I take advantage of the compressed nature of the lens, but only in the periphery, allowing users a wider field of view without sacrificing binocular vision. Experiments show that users can effectively see up to 180 degrees, and that the larger the object, the smaller the difference between the display and the naked eye in terms of visual acuity. The development of this display led to the design of a more adaptive and flexible framework, which I call ModulAR. The ModulAR display was developed to improve usability, configurability, and hands-free control of video see-through vision augmentation functionality. I utilize a prototype eye tracking system that has been integrated directly into an immersive display's frame to allow users to engage various vision augmentations in real time. I then develop a variety of visualizations designed to merge different fields of view, which are then refined and tested, revealing many interesting performance tendencies and generating significant qualitative feedback about the display. The method for classifying eye movements for engagement proves to be robust and re-affixing or re-configuring modules can be accomplished with minimal re-calibration. These methods be used to expand a user's range of fields of view, serve as a cornerstone for the development and study of new peripheral spatial compression functions and applications, and can allow for more flexible and on-demand configurations of future displays.

I hope that these works will inspire other researchers to develop better user centric content management algorithms and to build, share, and replicate new and interesting augmentations for human vision in the future.

Acknowledgements

This thesis would not have been possible without of the support of my mother. I will be forever grateful for her love and encouragement.

I would also especially like to thank my primary advisor, Kiyoshi Kiyokawa, for allowing me to study and grow in a flexible and positive educational environment. His guidance throughout the duration of my PhD has been essential to my success. Many thanks to professor Haruo Takemura for ensuring that funding was always available, and for encouraging rest and relaxation when it was most needed. I would also like to thank Daniel Sonntag for encouraging collaborations and facilitating guest research in Kaiserslautern and Saarbrucken, Germany. Thank you to professors Kuroda and Mashita for their guidance and help on a variety of topics. One more thank you goes out to Takumi Toyama, who has been a most excellent collaborator and friend over the past few years.

Another thank you goes out to the remaining staff and students at Takemura Lab for their guidance, companionship, and support. The efforts of the staff at the Graduate School of Information Science and Technology are also greatly appreciated. The sum of all parts truly made the experience a good one.

I also owe many thanks to the organizations who contributed to collaborative research efforts and funding, including Osaka University, the German Research Center for Artificial Intelligence (DFKI), Georgia Regents University, SensoMotoric Instruments (SMI), and to the Japan Society for the Promotion of Science (JSPS).

Research conducted during the duration of this PhD was funded in part by Grant-in-Aids for Scientific Research (B), #24300048 and #A15J030230 from the Japan society for the Promotion of Science (JSPS) and by the Kognit Project, which is supported by the German Federal Ministry of Education and Research (BMBF)

Lastly, I would also like to express another sincere thank you to the countless experiment participants who voluntarily participated and did their best in the tedious evaluations and experiments. Your time is greatly valued.

This thesis is dedicated to Jolie.

Bibliography

Ajanki, A., Billinghurst, M., Gamper, H., Järvenpää, T., Kandemir, M., Kaski, S., & Tossavainen, T. An augmented reality interface to contextual information. In *Virtual Reality, 15(2-3),* pp. 161–173, 2011.

Akeley, K., Watt, S. J., Girshick, A. R., & Banks, M. S. A stereo display prototype with multiple focal distances. In *ACM Transactions on Graphics (TOG) (23, No. 3)*, pp. 804–813, 2004.

Ames, S. L., & McBrien, N. A. Development of a miniaturized system for monitoring vergence during viewing of stereoscopic imagery using a head-mounted display. In *Proceedings of SPIE (5291),* pp. 25–35, 2004.

Ardouin, J., Lécuyer, A., Marchal, M., Riant, C., & Marchand, E. FlyVIZ: a novel display device to provide humans with 360 vision by coupling catadioptric camera with hmd. In *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology (VRST),* pp. 41–44, 2012.

Ashmore, M., Duchowski, A. T., & Shoemaker, G. Efficient eye pointing with a fisheye lens. *In Proceedings of Graphics Interface (GI),* pp. 203–210, 2005.

Avery, B., Sandor, C., & Thomas, B. H. Improving spatial perception for augmented reality x-ray vision. In *Proceedings of the Virtual Reality Conference,* pp. 79–82, 2009.

Azuma, R., & Furmanski, C. Evaluating label placement for augmented reality view management. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR),* p. 66, 2003.

Bar-Cohen, Y. Haptic devices for virtual reality, telepresence, and human-assistive robotics. In *Biologically Inspired Intelligent Robots, 122,* p. 73. 2003.

Belimpasakis, P., Roimela, K., & You, Y. Experience explorer: a life-logging platform based on mobile context collection. In *Proceedings of the Third International Conference on Next Generation Mobile Applications, Services and Technologies,* pp. 77–82, 2009.

Bell, B., Feiner, S., & Höllerer, T. View management for virtual and augmented reality. In *Proceedings of the 14th annual ACM symposium on User interface Software and Technology (UIST),* pp. 101–110, 2001.

Belongie, S., Malik, J., & Puzicha, J. Shape matching and object recognition using shape contexts. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 24(4),* pp. 509–522, 2002.

Berning, M., Yonezawa, T., Riedel, T., Nakazawa, J., Beigl, M., & Tokuda, H. pARnorama: 360 degree interactive video for augmented reality prototyping. In *Proceedings of the ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, pp. 1471–1474, 2013.

Billinghurst, M., Bowskill, J., Jessop, M., & Morphett, J. A wearable spatial conferencing space. In *Proceedings of the Second International Symposium on Wearable Computers (ISWC), Digest of Papers*, pp. 76–83, 1998.

Billinghurst, M., Kato, H., & Poupyrev, I. The magicbook-moving seamlessly between reality and virtuality. In *IEEE Computer Graphics and Applications, 21(3),* pp. 6–8, 2001.

Bimber, O., Wetzstein, G., Emmerling, A., & Nitschke, C. Enabling view-dependent stereoscopic projection in real environments. In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 14–23, 2005.

Birkfellner, W., Figl, M., Huber, K., Watzinger, F., Wanschitz, F., Hummel, J., & Bergmann, H. A head-mounted operating binocular for augmented reality visualization in medicine-design and initial evaluation. In *IEEE Transactions on Medical Imaging, 21(8),* pp 991–997, 2002.

Bitsch Link, J., Gerdsmeier, A., Smith P., F, & Wehrle, K. Indoor navigation on wheels (and on foot) using smartphones. In *Proceedings of Indoor Positioning and Indoor Navigation (IPIN),* p. 15, 2012.

Blaha, J., & Gupta, M. *Diplopia: A virtual reality game designed to help amblyopics.* In *Proceedings of IEEE Virtual Reality (VR),* pp. 163–164, 2014.

Bradski, G., & Kaehler, A. Learning OpenCV: Computer vision with the OpenCV library, O'Reilly Media, Inc., 2008.

Brandt, T., Dichgans, J., & Koenig, E. Differential effects of central versus peripheral vision on egocentric and exocentric motion perception. In *Experimental Brain Research, 16(5),* pp. 476–491, 1973.

Bulling, A., & Gellersen, H. Toward mobile eye-based human-computer interaction. In *IEEE Pervasive Computing, 9(4),* pp. 8–12, 2010.

Chang, C. C., & Lin, C. J. LIBSVM: a library for support vector machines. In *ACM Transactions on Intelligent Systems and Technology (TIST), 2(3),* p 27, 2011.

Chaudhri, V. K., Lane, H. C., Gunning, D. & Roschelle, J. Intelligent Learning Technologies Part 2: Applications of Artificial Intelligence to Contemporary and Emerging Educational Challenges. In *AI Magazine 34(4),* pp. 10–12, 2013.

Chen, C. B. Wide field of view, wide spectral band off-axis helmet-mounted display optical design. In *Proceedings of International Optical Design Conference,* pp. 61–66, 2002.

Chen, J., Pyla, P. S., & Bowman, D. Testbed evaluation of navigation and text display techniques in an information-rich virtual environment. In *Proceedings of IEEE Virtual Reality (VR),* pp. 181–188, p. 289, 2004.

Cho, I., Dou, W., Wartell, Z., Ribarsky, W., & Wang, X. Evaluating depth perception of volumetric data in semi-immersive VR. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 266–269, 2012.

Clarkson, E., Clawson, J., Lyons, K., & Starner, T. An empirical study of typing rates on mini-QWERTY keyboards. In *Extended Abstracts on Human Factors in Computing Systems*, pp. 1288–1291, 2005.

Coskun, T., Artinger, E., Pirritano, L., Korhammer, D., Benzina, A., Grill, C., & Klinker, G.. Gestyboard: A 10-finger-system and gesture based text input system for multi-touchscreens with no need for tactile feedback. In *Space, 5,* p. F6, 2012.

Dommes, A., & Cavallo, V. The beneficial effects of a simulator-based training on the elderly pedestrian safety. In *Proceedings of 12th International Conference on Mobility and Transport for Elderly and Disabled Persons*, p. 10, 2010.

Draper, M. H., Viirre, E. S., Furness, T. A., & Gawron, V. J. Effects of image scale and system time delay on simulator sickness within head-coupled virtual environments. In *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *43(1),* pp. 129–146, 2001.

Edelson, Danoff. Walking on an Electric Treadmill While Performing VDT Office Work. In *SIGCHI Bulletin, July 1989, 21(1),* pp. 72–77, 1989.

Ens, B. M., Finnegan, R., & Irani, P. P. The personal cockpit: a spatial interface for effective task switching on head-worn displays. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*, pp. 3171–3180, 2014.

Extron. <http://www.extron.com/img/mktg/environconhumanfact_fig2_3.jpg>. Accessed September 9th, 2015.

Fan, K., Huber, J., Nanayakkara, S., & Inami, M. SpiderVision: extending the human field of view for augmented awareness. In *Proceedings of the 5th Augmented Human (AH) International Conference,* No. 49, 2014.

Fischer, C., Muthukrishnan, K., Hazas, M., & Gellersen, H. Ultrasound-aided pedestrian dead reckoning for indoor navigation. In *Proceedings of the first ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, pp. 31–36, 2008.

Fukumoto, M., & Tonomura, Y. Body coupled FingerRing: wireless wearable keyboard. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems,* pp. 147–154, 1997.

Gabbard, JL., Swan II, Hix, D., Schulman, RS., Lucas, J., Gupta, D. An Empirical User-Based Study of Text Drawing Style and Outdoor Background Textures for Augmented Reality. In *Proceedings of IEEE Virtual Reality (VR) Technical Papers,* pp. 11–18, 2005.

Gattullo, M., Uva, A. E., Fiorentino, M., & Monno, G. Effect of Text Outline and Contrast Polarity on AR Text Readability in Industrial Lighting. In *IEEE Transactions on Visualization and Computer Graphics (TVCG), 21*(5), pp. 638–651, 2015.

Gemmell, J., Bell, G., Lueder, R., Drucker, S., & Wong, C. MyLifeBits: fulfilling the Memex vision. In *Proceedings of the tenth ACM International Conference on Multimedia,* pp. 235–238, 2002.

Green, N., Kruger, J., Faldu, C., & St Amant, R. A reduced QWERTY keyboard for mobile text entry. In *Extended Abstracts on Human Factors in Computing Systems*, pp. 1429–1432, 2004.

Gu, Y., & Sosnovsky, S. Recognition of student intentions in a virtual reality training environment. In *Proceedings of the Companion Publication of the 19th International Conference on Intelligent User Interfaces (IUI),* pp. 69–72, 2014.

Guerrero, L. A., Vasquez, F., & Ochoa, S. F. An indoor navigation system for the visually impaired. In *Sensors*, *12(6),* pp. 8236–8258, 2012.

Gupta, S. K., Anand, D. K., Brough, J., Schwartz, M., & Kavetsky, R. Training in Virtual Environments. A Safe, cost-effective, and engaging approach to training. *CALCE EPSC Press,* 2008.

Harper, R., Culham, L., & Dickinson, C. Head mounted video magnification devices for low vision rehabilitation: a comparison with existing technology. In *British Journal of Ophthalmology, 83(4),* pp. 495–500, 1999.

Hartmann, B., Klemmer, S. R., Bernstein, M., Abdulla, L., Burr, B., Robinson-Mosher, A., & Gee, J. Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of the 19th annual ACM Symposium on User Interface Software and Technology (UIST),* pp. 299–308, 2006.

Hartmann, K., Ali, K., & Strothotte, T. Floating labels: Applying dynamic potential fields for label layout. In *Smart Graphics,* pp. 101–113, 2004.

Häuslschmid, R., Osterwald, S., Lang, M., & Butz, A. Augmenting the Driver's View with Peripheral Information on a Windshield Display. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI),* pp. 311–321, 2015.

Haverinen, J., & Kemppainen, A. Global indoor self-localization based on the ambient magnetic field. In *Robotics and Autonomous Systems, 57(10),* pp. 1028–1035, 2009.

He, Z., & Yang, X. Hand-based interaction for object manipulation with augmented reality glasses. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry,* pp. 227–230, 2014.

Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., Smyth, G., Kapur, N., & Wood, K. SenseCam: A retrospective memory aid. In *Proceedings of Ubiquitous Computing,* pp. 177–193, 2006.

Hong, T. C., Kuan, N. A., Kiang, T. K., & John, S. K. T. Evaluation of Input Devices for Pointing, Dragging and Text Entry Tasks On A Tracked Vehicle. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 55(1*), pp. 2078–2082, 2011.

Hough, P.V.C. Methods and means for recognizing complex patterns, U.S. Patent 3 069 654, 1962.

Hua, H., Gao, C., Brown, L. D., Ahuja, N., & Rolland, J. P. A testbed for precise registration, natural occlusion and interaction in an augmented environment using a head-mounted projective display (HMPD). In *Proceedings of IEEE Virtual Reality (VR),* pp. 81–89, 2002.

Huckauf, A., Urbina, M. H., Grubert, J., Böckelmann, I., Doil, F., Schega, L., ... & Mecke, R. Perceptual issues in optical-see-through displays. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization,* pp. 41–48, 2010.

Hutter, M. Sequential predictions based on algorithmic complexity. In *Journal of Computer and System Sciences*, *72*(1), pp. 95–117. 2006.

Inoue, Y., Sashima, A., Ikeda, T., & Kurumatani, K. Indoor emergency evacuation service on autonomous navigation system using mobile phone. In *Universal Communication,* pp. 79–85, *2008.*

Irawati, S., Ahn, S., Kim, J., & Ko, H. Varu framework: Enabling rapid prototyping of VR, AR and ubiquitous applications. In *Proceedings of the IEEE Virtual Reality Conference,* pp. 201–208, 2008.

Ishiguro, Y., & Rekimoto, J. Peripheral vision annotation: noninterference information presentation method for mobile augmented reality. In *Proceedings of the 2nd Augmented Human (AH) International Conference,* p. 8, 2011.

Jankowski, J., Samp, K., Irzynska, I., Jozwowicz, M., & Decker, S. Integrating text with video and 3d graphics: The effects of text drawing styles on text readability. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1321–1330, 2010.

Jeni, L. A., Lőrincz, A., Szabó, Z., Cohn, J. F., & Kanade, T. Spatio-temporal Event Classification using Time-series Kernel based Structured Sparsity. In *Computer Vision–ECCV*, pp. 135–150, 2014.

Ki, J., Kwon, Y. M., & Sohn, K. 3D Gaze Tracking and Analysis for Attentive Human Computer Interaction. In *Proceedings of Frontiers in the Convergence of Bioscience and Information Technologies*, pp. 617–621, 2007.

Kim, H. D., Seo, S. W., Jang, I. H., & Sim, K. B. SLAM of mobile robot in the indoor environment with digital magnetic compass and ultrasonic sensors. In *Proceedings of the International Conference on Control, Automation and Systems*, pp. 87–90, 2007.

Kim, S. K., Kim, D. W., Kwon, Y. M., & Son, J. Y. Evaluation of the monocular depth cue in 3D displays. In *Optics Express, 16(26),* pp. 21415–21422, 2008.

Kim, S. K., Kim, E. H., & Kim, D. W. Full parallax multifocus three-dimensional display using a slanted light source array. In *Optical Engineering, 50(11),* pp. 114001-1–114001-4, 2011.

Kishishita, N., Orlosky, J., Mashita, T., Kiyokawa, K., & Takemura, H. Investigation on the peripheral visual field for information display with real and virtual wide field-of-view see-through HMDs. In *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI),* pp. 143–144, 2013.

Kiyokawa, K., Billinghurst, M., Campbell, B., & Woods, E. An occlusion-capable optical see-through head mount display for supporting co-located collaboration. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR),* p. 133, 2003.

Kiyokawa, K. A wide field-of-view head mounted projective display using hyperbolic half-silvered mirrors. In *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR),* pp. 1–4, 2007.

Kiyokawa, K., Takemura, H., & Yokoya, N. SeamlessDesign for 3D object creation. In *IEEE Multimedia*, 7(1), pp. 22–33, 2000.

Klann, M. Playing with fire: user-centered design of wearable computing for emergency response. In *Mobile Response*, pp. 116–125, 2007.

Klann, M., Riedel, T., Gellersen, H., Fischer, C., Oppenheim, M., Lukowicz, P., Pirkl, G. Lifenet: an ad-hoc sensor network and wearable system to provide firefighters with navigation support. In *the Adjunct Proceedings of UbiComp,* pp. 124–127, 2007.

Kruijff, E., Swan II, J. E., & Feiner, S. Perceptual issues in augmented reality revisited. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR),* pp. 3–12, 2010.

Kwon, Y. M., Jeon, K. W., Ki, J., Shahab, Q. M., Jo, S., & Kim, S. K. 3D Gaze Estimation and Interaction to Stereo Display. In *The International Journal of Virtual Reality (IJVR), 5(3),* pp. 41–45, 2006.

Lang, C., Nguyen, T. V., Katti, H., Yadati, K., Kankanhalli, M., & Yan, S. Depth matters: Influence of depth cues on visual saliency. In *Computer Vision–ECCV,* pp. 101–115, 2012.

Le, H., Dang, T., & Liu, F. Towards long-term large-scale visual health monitoring using Cyber Glasses. In *Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare,* pp. 200–207, 2013.

Lee, J. Y., Lee, S. H., Park, H. M., Lee, S. K., Choi, J. S., & Kwon, J. S. Design and implementation of a wearable AR annotation system using gaze interaction. In *Digest of Technical Papers International Conference on Consumer Electronics (ICCE),* pp. 185–186, 2010.

Leykin, A., & Tuceryan, M. Automatic determination of text readability over textured backgrounds for augmented reality systems. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 224–230, 2004.

Li, S. Binocular spherical stereo. In *IEEE Transactions on Intelligent Transportation Systems, 9(4),* pp. 589–600, 2008.

Lintu, A., & Magnor, M. An augmented reality system for astronomical observations. In *Proceedings of the IEEE Virtual Reality Conference,* pp. 119–126, 2006.

Liu, S., & Hua, H. A systematic method for designing depth-fused multi-focal plane three-dimensional displays. In *Optics Express, 18(11),* pp. 11562–11573, 2010.

Liu, S., Hua, H., & Cheng, D. A novel prototype for an optical see-through head-mounted display with addressable focus cues. In *IEEE Transactions on Visualization and Computer Graphics (TVCG), 16(3),* pp. 381–393, 2010.

Loomis, J. M., Kelly, J. W., Pusch, M., Bailenson, J. N., & Beall, A. C. Psychophysics of perceiving eye-gaze and head direction with peripheral vision: Implications for the dynamics of eye-gaze behavior. In *Perception, 37(9),* pp. 1443–1457, 2008.

Looser, J., Billinghurst, M., & Cockburn, A. Through the looking glass: the use of lenses as an interface tool for Augmented Reality interfaces. In *Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia,* pp. 204–211, 2004.

Lowe, D. G. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE International Conference on Computer Vision (ICCV),* pp. 1150–1157, 1999.

Lu, H., Zhao, J., Li, X., & Li, J. A new method of double electric compass for localization. In *Proceedings of the Sixth World Congress on Intelligent Control and Automation,* Vol. 2, pp. 5277–5281, 2007.

Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., & Looney, E. W. Twiddler typing: one-handed chording text entry for mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* pp. 671–678, 2004.

Maass, S., Döllner, J. Dynamic Annotation of Interactive Environments using Object-Integrated Billboards. In *Proceedings of the 2006 International Conference on Computer Graphics, Visualization and Computer Vision,* pp. 327–334, 2006.

Maglio, P. P., Barrett, R., Campbell, C. S., & Selker, T. SUITOR: An attentive information system. In *Proceedings of the 5th International Conference on Intelligent User Interfaces (IUI),* pp. 169–176, 2000.

Maimone, A., & Fuchs, H. Computational augmented reality eyeglasses. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR),* pp. 29–38, 2013.

Makita, K., Kanbara, M., & Yokoya, N. View management of annotations for wearable augmented reality. In *Proceedings of the IEEE International Conference on Multimedia and Expo,* pp. 982–985, 2009.

Mann, S., Fung, J., Aimone, C., Sehgal, A., & Chen, D. Designing EyeTap digital eyeglasses for continuous lifelong capture and sharing of personal experiences. In *Proceedings of Alt. Chi, the SIGCHI Conference on Human Factors in Computing Systems, 2005.*

Martin-Gonzalez, A., Heining, S. M., & Navab, N. Sight-based Magnification System for Surgical Applications. In *Bildverarbeitung für die Medizin,* pp. 26–30, 2010.

Matias, E., MacKenzie, I. S., & Buxton, W. Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY. In *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems,* pp. 88–94, 1993.

Maus, H., Schwarz, S., & Dengel, A. Weaving Personal Knowledge Spaces into Office Applications. In *Integration of Practice-Oriented Knowledge Technology: Trends and Prospectives*, pp. 71–82, 2013.

Mendez, E., Kalkofen, D., & Schmalstieg, D. Interactive context-driven visualization tools for augmented reality. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 209–218, 2006.

Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. Augmented reality: A class of displays on the reality-virtuality continuum. In *Photonics for Industrial Applications*, pp. 282–292, 1995.

Mirza, R., & Tehseen, A. An indoor navigation approach to aid the physically disabled people. In *Proceedings of the International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, pp. 979–983, 2012.

Montemerlo, M., Pineau, J., Roy, N., Thrun, S., & Verma, V. Experiences with a mobile robotic guide for the elderly. In *Proceedings of AAAI Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pp. 587–592, 2002.

Mon-Williams, M., Wann, J.P., & Rushton, S. Binocular Vision in a Virtual World: Visual Deficits Following the Wearing of a Head-Mounted Display, In *Ophthalmic and Physiological Optics, 13(4)*, pp. 387–391, 1993.

Mulloni, A., Dünser, A., & Schmalstieg, D. Zooming interfaces for augmented reality browsers. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 161–170, 2010.

Nagahara, H., Yagi, Y., & Yachida, M. A wide‐field‐of‐view catadioptrical head‐mounted display. In *Electronics and Communications in Japan (Part II: Electronics), 89(9)*, pp. 33–43, 2006.

Navarro, D., & Benet, G. Magnetic map building for mobile robot localization purpose. In *Proceedings of the IEEE Conference on Emerging Technologies & Factory Automation*, pp. 1–4, 2009.

Nelson, T.W., Roe, M., Bolia, R., Morley, R. Assessing Simulator Sickness in a See-through HMD. In *Proceedings of the Image 2000 Conference*, 2000.

Orlosky, J., Kiyokawa, K., & Takemura, H. Dynamic text management for see-through wearable and heads-up display systems. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI)*, pp. 363–370, 2013.

Orlosky, J., Toyama, T., Sonntag, D., Sarkany, A., & Lorincz, A. On-body multi-input indoor localization for dynamic emergency scenarios: fusion of magnetic tracking and optical character recognition with mixed-reality display. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 320–325, 2014.

Orlosky, J., Wu, Q., Kiyokawa, K., Takemura, H., & Nitschke, C. Fisheye vision: peripheral spatial compression for improved field of view in head mounted displays. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction (SUI)*, pp. 54–61, 2014.

Orlosky, J., Toyama, T., Kiyokawa, K., & Sonntag, D. ModulAR: Eye-controlled Vision Augmentations for Head Mounted Displays. In *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 21(11), 1259–1268, 2015.

Oskiper, T., Sizintsev, M., Branzoi, V., Samarasekera, S., & Kumar, R. Augmented Reality binoculars. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 219–228, 2013.

Ovrvision. http://www.ovrvision.com. Accessed February 18th, 2015.

Paletta, L., Santner, K., & Fritz, G. An Integrated System for 3D Gaze Recovery and Semantic Analysis of Human Attention. arXiv preprint arXiv:1307.7848, 2013.

Pan, M. S., Tsai, C. H., & Tseng, Y. C. Emergency guiding and monitoring applications in indoor 3D environments by wireless sensor networks. In *International Journal of Sensor Networks, 1(1),* pp. 2–10, 2006.

Pardhan, S., & Whitaker, A. Binocular summation in the fovea and peripheral field of anisometropic amblyopes. In *Current Eye Research, 20(1),* pp. 35–44, 2000.

Park, H. M., Lee, S. H., & Choi, J. S. Wearable augmented reality system using gaze interaction. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 175–176, 2008.

Petersen, N., Pagani, A., & Stricker, D. 2013, October. Real-time modeling and tracking manual workflows from first-person vision. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 117–124, 2013.

Pavlovych, A., & Stuerzlinger, W. Model for non-expert text entry speed on 12-button phone keypads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 351–358, 2004.

Percer, J. Child pedestrian safety education: Applying learning and developmental theories to develop safe street-crossing behaviors, In *Report No. HS-811 190, the U.S. Department of Transportation,* 2009.

Petersen, N., & Stricker, D. Learning task structure from video examples for workflow tracking and authoring. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2012,* pp. 237–246, 2012.

Plemmons, D., & Holz, D. Creating next-gen 3D interactive apps with motion control and Unity3D. In *Proceedings of ACM SIGGRAPH 2014 Studio,* p. 24, 2014.

Pollack, M. E., Brown, L., Colbry, D., McCarthy, C. E., Orosz, C., Peintner, B., Ramakrishnan, S., & Tsamardinos, I. Autominder: An intelligent cognitive orthotic system for people with memory impairment. In *Robotics and Autonomous Systems, 44(3)*, pp. 273–282, 2003.

Prothero, J. D., & Hoffman, H. G. Widening the field-of-view increases the sense of presence in immersive virtual environments. In *Human Interface Technology Laboratory Technical Report* TR-95, 2, 2005.

Ravi, N., Dandekar, N., Mysore, P., & Littman, M. L. Activity recognition from accelerometer data. In *Proceedings of AAAI Conference on Innovative Applications of Artificial Intelligence (IAAI),* pp. 1541–1546, 2005.

Reitmayr, G., Eade, E., & Drummond, T. W. Semi-automatic annotations in unknown environments. In *Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 67–70, 2007.

Rekimoto, J. The magnifying glass approach to augmented reality systems. In Proceedings of *International Conference on Artificial Reality and Tele-Existence (ICAT),* 95, pp. 123–132, 1995.

Reshko, G., Mason, M. T., & Nourbakhsh, I. R. Rapid prototyping of small robots. *Technical Report CMU-RI-TR-02-11, Robotics Institute, Carnegie Mellon University,* 2002.

Rolland, J. P., & Fuchs, H. Optical versus video see-through head-mounted displays in medical visualization. In *Presence: Teleoperators and Virtual Environments*, *9*(3), pp. 287–309, 2000.

Rueppel, U., & Stuebbe, K. M. BIM-based indoor-emergency-navigation-system for complex buildings. In *Tsinghua Science & Technology, 13(S1)*, pp. 362–367, 2008.

Scharff, L., Hill, A., & Ahumada, A. Discriminability measures for predicting readability of text on textured backgrounds. In *Optics Express*, *6*(4), pp. 81–91, 2000.

Schick, A., Morlock, D., Amma, C., Schultz, T., & Stiefelhagen, R. Vision-based handwriting recognition for unrestricted text input in mid-air. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, pp. 217–220, 2012.

Schowengerdt, B. T., & Seibel, E. J. True three-dimensional displays that allow viewers to dynamically shift accommodation, bringing objects displayed at different viewing distances into and out of focus. In *CyberPsychology & Behavior, 7*(6), pp. 610–620, 2004.

Scott, J., Izadi, S., Rezai, L. S., Ruszkowski, D., Bi, X., & Balakrishnan, R. RearType: text entry using keys on the back of a device. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 171–180, 2010.

Sellen, A. J., Fogg, A., Aitken, M., Hodges, S., Rother, C., & Wood, K. Do life-logging technologies support memory for the past?: an experimental study using sensecam. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 81–90, 2007.

Serina, E.R., Tal, R. and Rempel, D., Wrist and forearm postures and motions during typing. In *Ergonomics*, *42(7)*, pp. 938–951, 1999.

Sherstyuk, A., Treskunov, A., & Gavrilova, M. Predator-prey vision metaphor for multi-tasking virtual environments. In *Proceedings of IEEE Symposium on 3D User Interfaces (3DUI),* pp. 81–84, 2012.

Shum, H. Y., Kang, S. B., & Chan, S. C. Survey of image-based representations and compression techniques. *IEEE Transactions on Circuits and Systems for Video Technology, 13(11)*, pp. 1020–1037, 2003.

Sonntag, D., & Toyama, T. Vision-Based Location-Awareness in Augmented Reality Applications. In *Proceedings of the 3rd Workshop on Location Awareness for Mixed and Dual Reality (LAMDa)*, p. 5. 2013.

Sonntag, D., Zillner, S., Schulz, C., Weber, M., & Toyama, T. Towards Medical Cyber-Physical Systems: Multimodal Augmented Reality for Doctors and Knowledge Discovery about Patients. In *Design, User Experience, and Usability. User Experience in Novel Technological Environments, Lecture Notes in Computer Science Vol. 8014*, pp. 401–410, 2013.

Steed, A., & Julier, S. Behaviour-aware sensor fusion: Continuously inferring the alignment of coordinate systems from user behaviour. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR),* pp. 163–172, 2013.

Steptoe, W., Julier, S., & Steed, A. Presence and discernability in conventional and non-photorealistic immersive augmented reality. In Proceedings of *IEEE International Symposium on Mixed and Augmented Reality (ISMAR),* pp. 213–218, 2014.

Sullivan, A. 58.3: A Solid ‐ state Multi ‐ planar Volumetric Display. In *SID Symposium Digest of Technical Papers*, *34(1)*, pp. 1531–1533, 2003.

Suzuki, S. Topological structural analysis of digitized binary images by border following. In *Computer Vision, Graphics, and Image Processing, 30(1)*, pp. 32-46, 1985.

Swan, J. E., Livingston, M. A., Smallman, H. S., Brown, D., Baillot, Y., Gabbard, J. L., & Hix, D. A perceptual matching technique for depth judgments in optical, see-through augmented reality. In *Proceedings of the Virtual Reality Conference,* pp. 19–26, 2006.

Szabó, Z., Póczos, B. & Lőrincz, A. Online group-structured dictionary learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 2865–2872, 2011.

Takagi, A., Yamazaki, S., Saito, Y., & Taniguchi, N. Development of a stereo video see-through HMD for AR systems. In *Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISMAR),* pp. 68–77, 2000.

Tan, D. S., & Czerwinski, M. Effects of visual separation and physical discontinuities when distributing information across multiple displays. In *Proceedings of Interact, 3*, pp. 252–255, 2003.

Tanaka, K., Kishino, Y., Miyamae, M., Terada, T., & Nishio, S. An information layout method for an optical see-through head mounted display focusing on the viewability. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR),* pp. 139–142, 2008.

Tatzgern, M., Kalkofen, D., Grasset, R., & Schmalstieg, D. Hedgehog labeling: View management techniques for external labels in 3D space. In *Proceedings of IEEE Virtual Reality (VR),* pp. 27–32, 2014.

Thanedar, V., Hollerer, T. Semi-automated Placement of Annotations in Videos. *Technical Report #2004-11, Department of Computer Science, University of California,* 2004.

Toyama, T., Dengel, A., Suzuki, W., & Kise, K. Wearable Reading Assist System: Augmented Reality Document Combining Document Retrieval and Eye Tracking. In *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR),* pp. 30–34, 2013.

Toyama, T., Kieninger, T., Shafait, F., & Dengel, A. Gaze guided object recognition using a head-mounted eye tracker. In *Proceedings of the Symposium on Eye Tracking Research and Applications,* pp. 91–98, 2012.

Toyama, T., Sonntag, D., Orlosky, J., & Kiyokawa, K. Attention Engagement and Cognitive State Analysis for Augmented Reality Text Display Functions. In *Proceedings of the 20th International Conference on Intelligent User Interfaces (IUI)*, pp. 322–332, 2015.

Tsai, H. C. Safety view management for augmented reality based on MapReduce strategy on multi-core processors. In *Proceedings of the 13th International Conference on ITS Telecommunications (ITST)*, pp. 151–156, 2013.

Tseng, Y. C., Pan, M. S., & Tsai, Y. Y. A distributed emergency navigation algorithm for wireless sensor networks. In *IEEE Computers, 39*(7), pp. 55–62, 2006.

Uratani, K., Machida, T., Kiyokawa, K., & Takemura, H. A study of depth visualization techniques for virtual annotations in augmented reality. In *Proceedings of IEEE Virtual Reality*, pp. 295–296, 2005.

Urey, H., Chellappan, K. V., Erden, E., & Surman, P. State of the art in stereoscopic and autostereoscopic displays. In *Proceedings of the IEEE, 99*(4), pp. 540–555, 2011.

Van Den Bergen, G. Proximity queries and penetration depth computation on 3d game objects. In *Proceedings of Game Developers Conference, 170*, 2001.

Van, D. N., Mashita, T., Kiyokawa, K., & Takemura, H. Subjective evaluations on perceptual depth of stereo image and effective field of view of a wide-view head mounted projective display with a semi-transparent retro-reflective screen. In *Proceedings of IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 327–328, 2012.

Vargas-Martin, F., & Peli, E. Augmented-view for restricted visual field: multiple device implementations. In *Optometry & Vision Science*, *79*(11), pp. 715–723, 2002.

Veas, E., Grasset, R., Kruijff, E., & Schmalstieg, D. Extended overview techniques for outdoor augmented reality. In *IEEE Transactions on Visualization and Computer Graphics (TVCG), 18*(4), pp. 565–572, 2012.

Veas, E., Mulloni, A., Kruijff, E., Regenbrecht, H., & Schmalstieg, D. Techniques for view transition in multi-camera outdoor environments. In *Proceedings of Graphics Interface (GI)*, pp. 193–200, 2010.

Vertegaal, R. Designing attentive interfaces. In *Proceedings of the ACM Symposium on Eye Tracking Research & Applications (ETRA)*, pp. 23–30, 2002.

Wabirama, S., & Hamamoto, K. A Geometric Model for Measuring Depth Perception in Immersive Virtual Environment. In *Proceeding of Asia Pacific Conference on Computer Human Interaction (APCHI), 2*, pp. 325–330, 2012.

Watson, B. A., Walker, N., & Hodges, L. F. A user study evaluating level of detail degradation in the periphery of head-mounted displays. In *Proceedings of Framework for Interactive Virtual Environments (FIVE) Conference*, pp. 203–212, 1995.

Whelan, M., Oxley, J., Charlton, J., D'Elia, A., & Muir, C. Child Pedestrians: Factors Associated with Ability to Cross Roads Safely and Development of a Training Package. In *Report #283, Accident Research Centre, Monash University,* 2008.

Wierzbicki, R. J., Tschoeppe, C., Ruf, T., & Garbas, J. U. EDIS-Emotion-Driven Interactive Systems. In *Proceedings of the 5th International Workshop on Semantic Ambient Media Experience (SAME),* pp. 59–68, 2012.

Wither, J., DiVerdi, S., Hollerer, T. Annotation in outdoor augmented reality. In *Computers & Graphics, 33(6),* pp. 679–689, 2009.

Wood, J. M., Collins, M. J., & Carkeet, A. Regional variations in binocular summation across the visual field. In *Ophthalmic and Physiological Optics*, *12*(1), pp. 46–51, 1992.

Woods, R. L., Fetchenheuer, I., Vargas ‐ Martín, F., & Peli, E. The impact of non ‐ immersive head ‐ mounted displays (HMDs) on the visual field. In *Journal of the Society for Information Display, 11*(1), pp. 191–198, 2003.

Xiong, Y., & Turkowski, K. Creating image-based VR using a self-calibrating fisheye lens. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 237–243, 1997.

Xuan, Y., Sengupta, R., & Fallah, Y. Making indoor maps with portable accelerometer and magnetometer. In *Proceedings of Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS),* pp. 1–7, 2010.

Yamazaki, S., Inoguchi, K., Saito, Y., Morishima, H., & Taniguchi, N. Thin wide-field-of-view HMD with free-form-surface prism and applications. In *Electronic Imaging'99,* pp. 453–462, 1999.

Yano, Y., Kiyokawa, K., Sherstyuk, A., Mashita, T., Takemura, H. Investigation of Dynamic View Expansion for Head-Mounted Displays with Head Tracking in Virtual Environments. In *Proceedings of the 24th International Conference on Artificial Reality and Telexistence (ICAT)*, 2014.

Zhang, F., & Sun, H. Dynamic labeling management in virtual and augmented environments. In *Proceedings of the Ninth International Conference on Computer Aided Design and Computer Graphics*, p. 6. 2005.

Zokai, S., Esteve, J., Genc, Y., & Navab, N. (Multiview paraperspective projection model for diminished reality. In *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 217–226, 2003.