Spatial Analytic Interfaces

By

Barrett Ens

A thesis submitted to the Faculty of Graduate Studies of

The University of Manitoba

in partial fulfilment of the requirements of the degree of

Doctor of Philosophy

Department of Computer Science

University of Manitoba

Winnipeg

This dissertation was reviewed and approved by the following committee members:

Pourang Irani
Professor of Computer Science, University of Manitoba
Thesis Advisor

Neil Bruce
Assistant Professor of Computer Science, University of Manitoba

Bertram Unger
Research Director of the Clinical Learning and Simulation Facility, and
Assistant Professor of Internal Medicine, University of Manitoba

Ravin Balakrishnan
Department Chair and Professor of Computer Science, University of Toronto

Acknowledgements

I owe many thanks to my advisor, Pourang Irani, for encouraging me to undertake this PhD thesis. He has always pushed me to strive to accomplish my best work in my research and other endeavours. I am also grateful to have a committee with a fine balance of expertise. I thank its members, Neil Bruce, Bertram Unger, and Ravin Balakrishnan for their excellent feedback during the thorough revisions of my final work, and for leaving me with thoughts to contemplate.

I am grateful for all I have learned from my many mentors and peers at the University of Manitoba and beyond, who contributed to the work underlying this dissertation, or have offered their feedback and advice. The input of these many collaborators has helped push this work to its very best.

Many thanks are also due to the administration and technical staff who keep the Department of Computer Science and its equipment running smoothly behind the scenes.

Last, I thank my partner Katy and other members of my family for offering their support throughout my journey.

Copyright Notices and Disclaimers

Sections of this thesis have been published in conference proceedings and journal publications, either previously or forthcoming at the time of publication. Permissions for these works to appear in this dissertation have been granted by their respective publishers. Following is a list of prior publications in which portions of this work appeared, organized by chapter.

Portions of Chapters 1 and 3

Barrett Ens and Pourang Irani. 2016. Spatial Analytic Interfaces: Spatial user interfaces for in-situ visual analytics. *IEEE Computer Graphics and Applications*. IEEE, in press. http://doi.org/10.1109/MCG.2016.38
** This paper was selected for presentation at the annual IEEE VIS conference in a special session aimed at fostering synergies between theoretical and applied researchers*

The Majority of Chapter 2

Barrett Ens, Juan David Hincapié-Ramos, and Pourang Irani. 2014. Ethereal Planes: A design framework for 2D information spaces in 3D mixed reality environments. *Proceedings of the 2nd ACM Symposium on Spatial User Interaction (SUI '14)*. ACM, 2–12. http://doi.org/10.1145/2659766.2659769

The Majority of Chapter 4

Barrett M. Ens, Rory Finnegan, and Pourang P. Irani. 2014. The Personal Cockpit: A spatial interface for effective task switching on head-worn displays. *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 3171–3180. http://doi.org/10.1145/2556288.2557058

Portions of Chapter 5 and 6

Barrett Ens, Eyal Ofek, Neil Bruce, and Pourang Irani. 2015. Spatial constancy of surface-embedded layouts across multiple environments. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction (SUI '15)*. ACM, 65-68. http://dx.doi.org/10.1145/2788940.2788954

Barrett Ens, Eyal Ofek, Neil Bruce, and Pourang Irani. 2016. Shared Façades: Surface-embedded layout management for ad-hoc collaboration using head-worn displays. *Collaboration Meets Interactive Surfaces and Spaces (CMISS) – Theory and Practice*. Craig Anslow, Pedro Campos, Joaquim Jorge, (Eds). Springer, in press.

Barrett Ens, Ahmad Byagowi, Teng Han, Juan David Hincapié-Ramos, and Pourang Irani. Combining ring input with hand tracking for precise, natural interaction with spatial analytic interfaces. In *Proceedings of the 4th ACM Symposium on Spatial User Interaction (SUI '16)*. ACM, in press.

Abstract

We propose the concept of spatial analytic interfaces (SAIs) as a tool for performing in-situ, everyday analytic tasks. Mobile computing is now ubiquitous and provides access to information at nearly any time or place. However, current mobile interfaces do not easily enable the type of sophisticated analytic tasks that are now well-supported by desktop computers. Conversely, desktop computers, with large available screen space to view multiple data visualizations, are not always available at the ideal time and place for a particular task. Spatial user interfaces, leveraging state-of-the-art miniature and wearable technologies, can potentially provide intuitive computer interfaces to deal with the complexity needed to support everyday analytic tasks. These interfaces can be implemented with versatile form factors that provide mobility for doing such taskwork *in-situ*, that is, at the ideal time and place.

We explore the design of spatial analytic interfaces for in-situ analytic tasks, that leverage the benefits of an upcoming generation of light-weight, see-through, head-worn displays. We propose how such a platform can meet the five primary design requirements for personal visual analytics: mobility, integration, interpretation, multiple views and interactivity. We begin with a design framework for spatial analytic interfaces based on a survey of existing designs of spatial user interfaces. We then explore how to best meet these requirements through a series of design concepts, user studies and prototype implementations. Our result is a holistic exploration of the spatial analytic concept on a head-worn display platform.

Table of Contents

List of Tables

List of Figures

1 Introduction

Personal computing devices are becoming smaller yet more powerful, allowing greater user mobility, increased capacity for collecting and storing personal data, and enhanced control for managing these data to benefit our everyday lives. A catalyst in this shift in computer usage is increased access to sensors and interfaces, which are becoming integrated with what we normally wear. As we have already witnessed computers moving from entire rooms to desktops to pocket-sized devices, we are now experiencing a continuing shift to wearable form factors such as smart watches and digital eyewear. This new generation of interactive information displays has great potential to enrich our lives. Unlike with current mobile technology, information from these devices can be ingested from a glance at the wrist or even a slight eye movement. Such always-available information access allows in-situ computing: access to situationally appropriate data at an ideal time and place. By providing wearable technology with suitable information-seeking interfaces we can make computing a natural and 'invisible' part of our daily activities.

The complexity of mobile computing interfaces has so far been limited by the small available space for input and display. For example, some common tasks performed on mobile devices include consumption tasks such as reading or viewing videos, mobile communication tasks such as sending or receiving short messages, and organizational tasks such as keeping a list of contacts or setting reminders. As wearable device interfaces continue to shrink, current design solutions are trending further toward simplicity; new interface paradigms (e.g. Google Glass [73], Android Wear [5]) are designed to

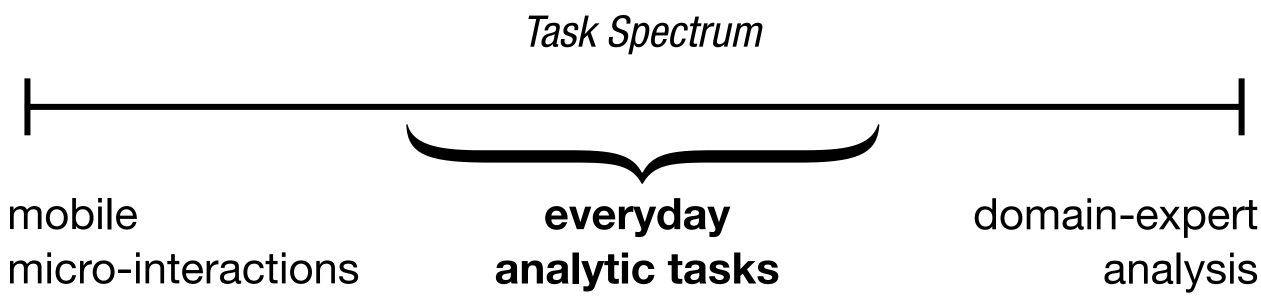

Figure 1. Task Spectrum
The nature of computer-assisted tasks varies widely from short-term, simple tasks carried out by mobile or wearable device users to intensive, analytic tasks carried out by teams of domain experts. We propose Spatial Analytic Interfaces (SAIs) for supporting everyday analytic tasks, which reside between the extremes on this spectrum.

support micro-interactions, short bursts of activity that avoid impinging on one's day-to-day activities by minimizing task duration.

In contrast to these current trends, we are interested in designing interfaces that support everyday analytic tasks (**Figure 1**). Properties of such tasks include the requirement for concerted thought, the integration of information from multiple sources, and the application of human sensemaking abilities (**Figure 3**). Typical examples of everyday analytic tasks include balancing a cheque book, planning a vacation itinerary, or conducting a price search for the best available deal on a particular item. Such tasks are commonly carried out with the assistance of computers, yet are not necessarily well supported by today's mobile device interfaces.

To design interfaces that support analytic tasks, we can draw from the field of visual analytics. Visual analytics is devoted to developing tools that help users gain insights through deep exploration of multiple interlinked visualizations of diverse data sets. Although originally aimed at supporting domain experts with intensive analysis, for instance of biomedical data [192] or military intelligence reports [191], visual analytic methods have been recently adopted for analysis of an increasing wealth of everyday personal information [100,168]. For example, sensors in people's homes track energy consumption and resource usage patterns; mobile computers such as smartphones and embedded automobile software continuously track owners' everyday movements; and wearable accessories are popular for tracking personal health and fitness data. This trend of ubiquitous data collection presents a growing need for tools to comprehend and digest the patterns of importance and to provide actionable results [53].

The benefits to be realized from an increasing prevalence of mobile and wearable technology are then twofold: While these devices allow the routine collection of useful activity data, they also provide an opportunity to facilitate in-situ data analysis. Homeowners concerned with minimizing their energy consumption, for instance, might be better able to make informed choices if appropriate information is available at the time when they are choosing how to consume resources or energy (e.g. turning up a thermostat). Similarly, if people are able to consult their banking history through a mobile app, they may make use of this information directly before making significant purchases. The mobile component is essential to in-situ computing, since waiting to view data at home on a personal computer results in the situational context becoming lost. However, viewing data on the small screen of a personal mobile device may be prohibitively cumbersome, and lacks the potential for gaining insight by controlling multiple, coordinated views of the data [191].

One promising approach to provide mobile interfaces for in-situ use, with advanced features to support analysis and sensemaking, is the application of spatial user interfaces. Spatial user interfaces leverage benefits such as spatial memory and proprioception to map information to physical space and have been shown to improve performance on some analytic tasks [56]. For instance, arranging multiple visualizations side-by-side can allow for faster and easier comparison than navigating between multiple components on a single abstract interface; the user can easily switch views using physical head or body motion and apply spatial memory to recall the location of important items, making for an efficient and intuitive experience. Several research studies have shown examples where interfaces that leverage motion in space over large displays [126,186], multiple displays [21,190] or through virtual navigation [38,121] can provide more efficient navigation or improved understanding of complex tasks.

We propose the concept of Spatial Analytic Interfaces (SAIs) as a solution for everyday data-monitoring and decision-making based on in-situ analysis. SAIs leverage the benefits of spatial user interfaces for completing in-situ, analytic tasks (**Figure 3**). The concept of SAIs is platform-agnostic, however we focus on head-worn display (HWD) technology as a particularly appropriate platform for meeting the requirements for supporting in-situ analytic taskwork. Such digital eyewear is currently available in lightweight form factors at an affordable cost for general consumers and the technology is rapidly advancing. HWDs are becoming equipped with depth cameras and inertial sensors that allow tracking of hand, fingertip and body motion (e.g. Meta, Microsoft HoloLens). These features will facilitate intuitive spatial interaction, for instance the ability to switch between spatially situated displays by turning one's head [192]. With robust spatial tracking, these devices essentially provide unlimited 'display' space; multiple information visualizations can be integrated directly into the appropriate home, work or mobile environment. Furthermore, virtual displays rendered by these wearable systems can be situated where they are most convenient for a given context, for instance on the kitchen counter or backsplash for monitoring home energy consumption, or in a hemispherical formation around the user's body in mobile situations such as shopping or jogging. This spatial paradigm can also support advanced techniques not possible with standard desktop displays; for example, visual links can span physical space to connect data across multiple displays or guide users to information that is not currently in their focus of attention [220].

The goal of this thesis is to introduce the concept of SAIs and examine the benefits this concept provides over current mobile interfaces. We define the requirements and challenges in implementing SAIs and undertake a broad exploration of design for such interfaces that address these requirements and challenges. We argue that spatial interfaces implemented on wearable platforms are capable of overcoming the limitations of current mobile technologies to provide computing tools for in-situ, analytic tasks.

We begin with a set of simple scenarios that demonstrate the benefits of spatial interfaces for in-situ analytic tasks, and then provide a set of requirements for a system that supports such tasks. Next, we conduct a survey of state-of-the-art spatial user interfaces, and make an informed choice to explore HWDs as a particular display and interaction platform for spatial analytics. Combined with compact input sensors such as depth cameras and inertial motion units, these wearable displays can support interactive interfaces that provide constant and convenient access to personal data. The remainder of the thesis follows a formal design process, exploring in depth how each of the fundamental requirements can be met through the design of a spatial HWD interface.

1.1 Scenarios

To demonstrate the breadth of potential opportunities for SAI, we discuss a number of scenarios where data visualizations presented on HWDs can be of potential value for in-situ analytic tasks. Within the scope of everyday analytic tasks, we discuss three specific categories, distributed across a range of the task spectrum (**Figure 2**): First, we discuss personal analytic tasks, which rest at the centre of this region of the spectrum. Next, we give some examples of ambient information displays that use SAI principles. Finally, we discuss how SAIs can be applied to more intensive analytic tasks for in-situ industrial applications.

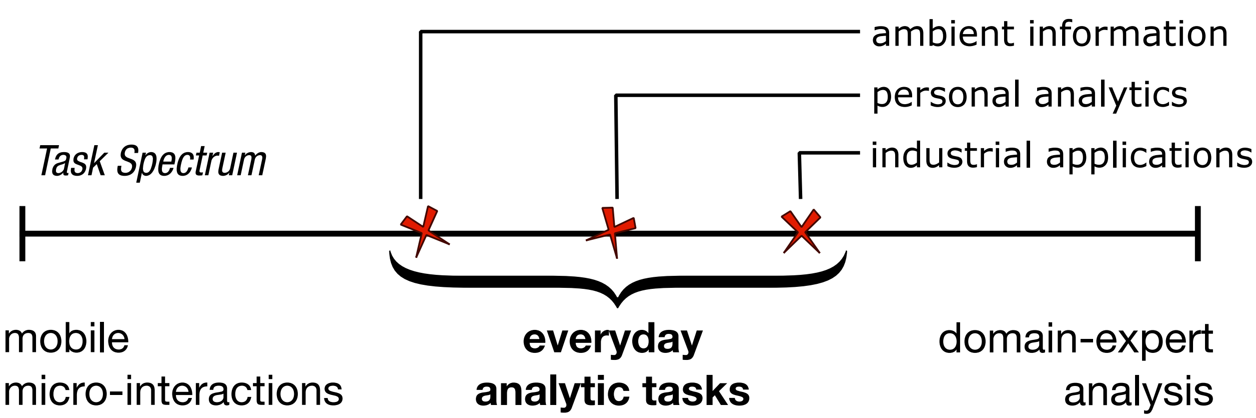

Figure 2 Categories of Everyday Analytic Tasks
Everyday analytic tasks include a range of activities, from gleaning information from ambient displays, to analysing personal data, to more intensive in-situ industrial applications.

Personal Analytic Task Scenarios

First we visit Ellie on her morning run (**Figure 3**a). Following her along are a pair of virtual display windows. The display to her left shows her step count, heart rate and estimated calories burned. The other, on her right, contains a map showing her current location and her predicted route, based on logs from previous runs. Neither window occludes her forward view; she periodically consults them by turning her head slightly to her left or right. Ellie pauses for a short break on a hilltop to drink some water and examine her progress. With a hand gesture, she makes the map window larger and places it at a

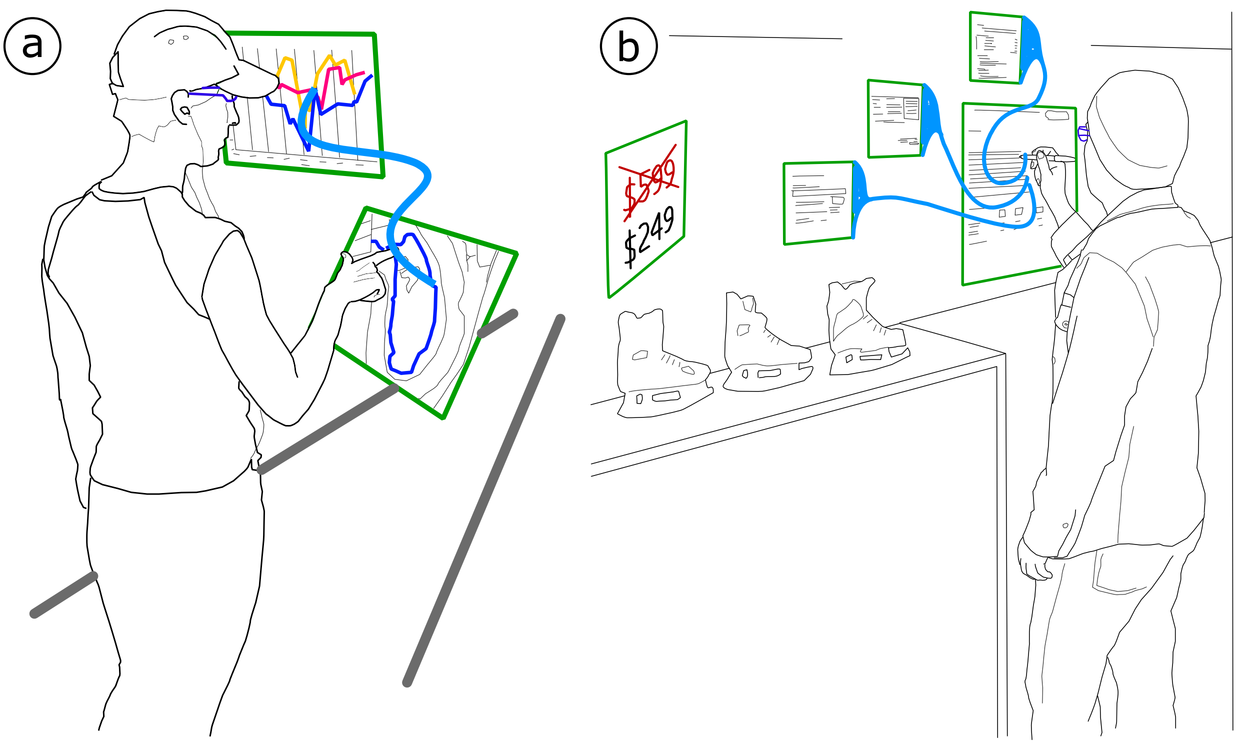

Figure 3 Personal Analytics Task Scenarios
At the center of the scope of everyday analytic tasks on the task spectrum (Figure 2) are personal analytic tasks. These tasks involve the in-situ exploration of personal data, such as examining the progress of a workout during a break (a), or reviewing one's expenses during a shopping excursion (b).

sloped angle at about waist level. At eye level, Ellie opens a new window showing a visualization of her heart rate, a graph with several different coloured lines representing the collected from pulse readings on her wrist band, with one line for each of the past few days. Sure enough, each of the lines has a peak at approximately the same time. Ellie 'taps' one of these peaks on the floating virtual display and then makes a gesture toward the map. A virtual link appears, connecting the high point on the graph to a spot on the marked path on the map. As suspected, the peak in the heart rate coincides with the location of a hill on Ellie's route.

This scenario exemplifies the type of task we envision SAI being useful. Foremost, it involves a typical problem that many people may encounter on a daily basis, in this case, analysing training data. The task is performed in-situ – during Ellie's workout – rather than later at home. Also, the task has analytic component, involving the exploration of data and linking of data points across multiple views. While this task can potentially be accomplished using a smartphone, we propose that a spatial interface will provide better support for switching between views, and potentially lead to a better understanding of the data and a more satisfying user experience.

This is but one of many example of everyday situations where SAIs may be used to assist in-situ analysis of personal data. As a second example, imagine a shopping excursion (**Figure 3**b), where Marcus, our imagined shopper, locates a tempting deal on a pair of ice skates. Rather than wait until he gets home to check his expenses, Markus pulls a stylus from his pocket and appropriates a nearby wall as a temporary workspace. With a few virtual strokes, he opens a spreadsheet and a pile of virtual bills. He spreads these on the surrounding wall space, where virtual links connect each bill to its corresponding line on the spreadsheet. Marcus is not concerned about the privacy of his information as the items are visible only to only him through his HWD.

This example shows the potential utility of SAIs for everyday tasks that we may not imagine as practical for in-situ performance. Making use of available space during in-situ experiences may support tasks that are too cumbersome to conduct on current mobile platforms, such as smartphones, and SAIs may potentially broaden the scope of in-situ computing activities to include those that are typically confined to desktops in home or office environments. This example also shows how HWD technologies, such as spatial sensors and see-thorough displays might be used to provide meta-information such as interspatial links, to take advantage of surrounding surfaces, and to handle concerns about information privacy.

Ambient Information Display Scenarios

To the left of personal analytic tasks on the task spectrum (**Figure 2**) are less intensive tasks that involve ambient information displays. These tasks are closer in nature to micro-interactions than the personal analytic task scenarios given above since they demand less attention. However, such monitoring or awareness tasks may nonetheless include an analytic component, where people make use of information visualizations to support in-situ decision making [100].

For instance, ambient information visualizations can be used to provide awareness of resource consumption in the home. Maintaining awareness is an important aspect of changing our behaviour, and making this information available through ambient displays can help to build and maintain this awareness [17]. Researchers have previously explored such ambient visualizations projected onto the wall of a shower stall [107] or kitchen backsplash [17], to provide relevant information about water or energy consumption.

As HWD technology continues to approach the form factor of current eyewear, these devices will become a convenient platform for the display of ambient information. As the HWD platform may be always worn, it is practical for in-situ use. For example, relevant information about resource consumption can be provided while adjusting the temperature of a room (**Figure 4**a), or while running water to do the dishes (**Figure 4**b), without the need to install projection equipment in each room. While these information displays need

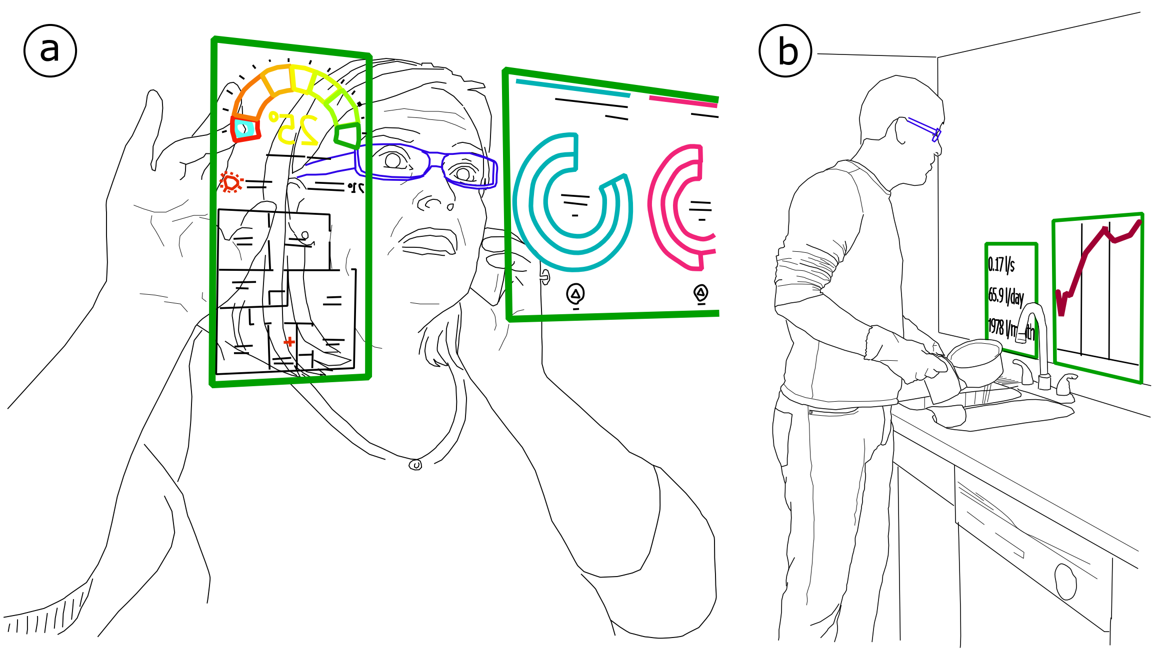

Figure 4. Ambient Display Scenarios
SAIs can take the form of ambient information displays that support in-situ decision making. For instance, awareness about energy consumption can be provided while adjusting the temperature of a room (a) or while running water to do the dishes (b).

be only glanced at occasionally, they can also provide opportunities for engaging in more intensive analytic tasks. For instance, if a home owner notices a spike in energy use while adjusting the heat, she may explore further to determine the average daily consumption of each room. Similarly, after doing a batch of dishes by hand, one may inquire whether less water was used than with a similar batch previously put in the dishwasher. These explorations of the available data may support decisions about how to improve future behaviours to meet target goals.

Industrial Application Scenarios

SAIs may also support industrial applications that are more analytically intensive than personal information visualization, occupying a space further to the right on the task spectrum (**Figure 2**). A spatial display layout viewed on a HWD platform would be particularly practical in situations where the hands are occupied, for instance for a medical practitioner who must keep her hands sterile (**Figure 5**a). During surgery, multiple displays can provide important information such as data from recent MRI or CT scans, and current vital signs. Tracking sensors embedded in the HWD allow the practitioner to manipulate these displays using mid-air gestures, without the need to touch any surface.

SAIs can also benefit mobile workers, for instance while carrying out inspection and maintenance activities in a large factory or processing plant (**Figure 5**b). Such tasks may benefit from available information about recent maintenance schedules or current status, such as sensor readings of flow and pressure in a factory pipeline. SAIs allow this

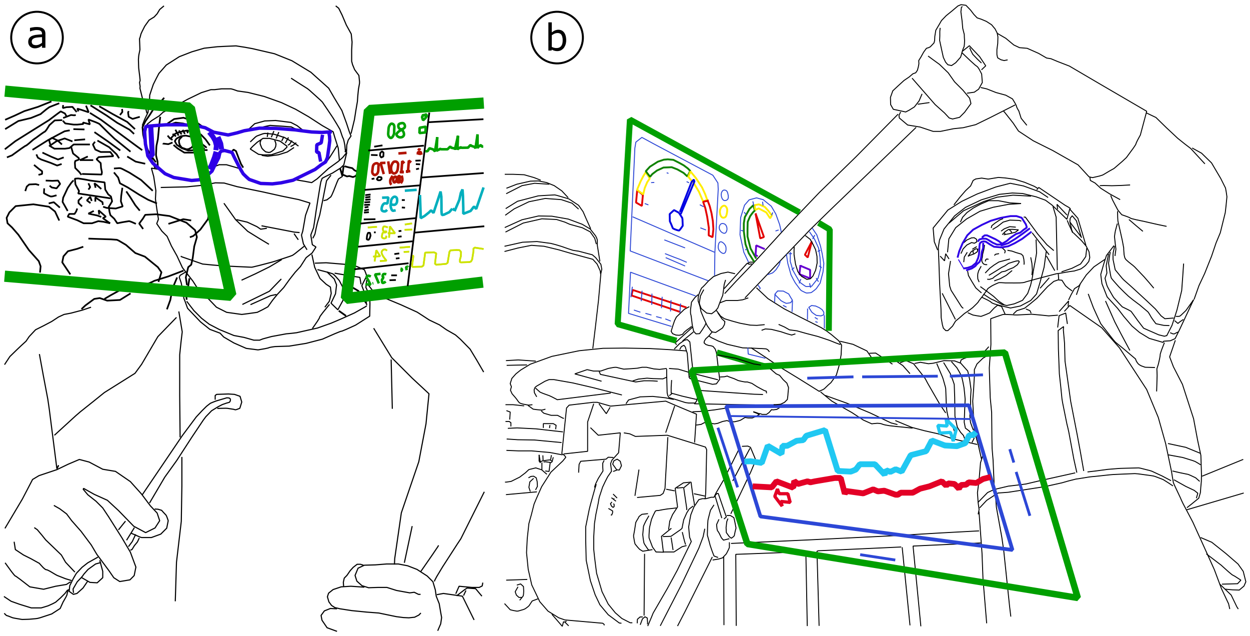

Figure 5. Industrial Application Scenarios.
SAIs can support intensive analytic tasks for industrial applications. For instance, a SAI can provide needed information for a surgeon (a), who can manipulate the spatial layout using mid-air gestures, without the need to touch a device surface. Information can also be distributed around a job site to provide important data in-situ to a mobile worker (b).

information to be provided in-situ, in essence taking the required information from the main control room and distributing it where needed as the worker moves around the job site. Information can be laid out across the physical equipment to support analytic sensemaking while on the go.

The above examples explore some opportunities that can be provided by SAIs on near-future HWDs. All illustrate typical, everyday activities that rely on analytic processes. This dissertation shows how our lives can be enriched by increasing the availability of information and the convenience of access using the in-situ visual analytic tools of an HWD, and by integrating these tools in our daily tasks.

1.2 The Five Requirements for Supporting In-Situ Analytic Tasks

To begin our discussion on what SAIs have to offer to the everyday user, we pose the following question: *As mobile and wearable technologies become an integral part of our everyday lives, what are the design requirements for an ideal platform to facilitate in-situ data analysis?* In response to this question, we propose a list of requirements, derived from several sources, including our own experience designing interactive systems; inspiration from the above scenarios; and, existing literature surveys on visual analytics. Among the latter seminal works are an exploration of interaction in visual analytic systems from Yi et al. [231], and an early look at adapting information visualization for everyday use by Pousman et al. [168]. More recently, a survey by Huang et al. [100] distils a general taxonomy for the design space of Personal Visual Analytics. From these and other relevant works we define a set of requirements specific to in-situ visual analytic tasks. This list contains five primary requirements: *Mobility*, *Integration*, *Interpretation*, *Multiple Views* and *Interactivity*. In the following descriptions of these items, we demonstrate how each builds upon the previous core concept.

Mobility

One implication of mobile devices is their ability to implicitly collect sensor data and infer activities of the user. This opportunity has been recently exploited by industry with the introduction of numerous tracking devices and has resulted in the recent 'quantified self' movement — the collection and analysis of one's own personal data — aimed at making use of this data [100], for example to benefit users' health. However, the activities of data collection and analysis are primarily conducted separately, for instance by periodic recommendations (e.g. a reminder to stand up every 30 minutes) or by more intensive analysis supported by desktop tools. In contrast, we posit that supporting in-situ analysis — allowing users to analyse data directly in the situations where they are applied — will help users gain the most benefit from their data. Based on their in-depth survey, Huang et al. [100] suggest that incorporating analysis tasks into users' daily activities can help encourage adoption of analysis tools. For instance, presenting data about commuting habits at the time of the activity [69] can help users make informed choices. Likewise, if a jogger (**Figure 3**a) wishes to track her heart rate and estimated calories burned for training purposes, she may benefit from the ability to monitor these data during a run. This would allow her to alter her physical activity levels immediately, in contrast to comparing daily records at home on a desktop computer. In many instances such access requires the analysis tools to be mobile and usable is a range of potential situations.

Integration

In addition to being embedded in mobile or wearable devices, sensors that collect data about our daily activities can be embedded in places frequented by users, such as homes and offices. Likewise, many potential scenarios for using analytic tools can be done in-situ in these environments. Similarly, another method proposed by Huang et al. [100] for encouraging user adoption of analytic tools is to integrate visualizations into the environment. By integrating visualizations into the surrounding environment, the visualizations become readily available to the user while interfering minimally with their task. For instance, a reminder about the costs of excess water consumption (**Figure 4**b) is most actionable if available when and where the water is being used, say on a vanity mirror to inform a homeowner about the cost of leaving the water running while shaving.

Interpretation

Whether in-situ analysis is conducted in a stationary or mobile context, the adoption of analytic tools will depend on their ease of use. Pousman et al. [168] made several recommendations for adopting visual analytic techniques to everyday situations. They suggest that visualizations should provide the most immediately relevant information, should present data in a form that is intuitive or easy to learn, and should be aesthetically pleasing to encourage contemplation. In relation to the above criteria, i.e. in the case of mobile scenarios or in those were visualizations are integrated into the environment, we add that the format of a particular visualization should be adapted to the given context; for example, information consumed in a mobile context should be highly simplified, while that integrated into a home appliance should fit both the physical form and use case of the appliance.

Multiple Views

In some contexts, sensemaking can be assisted by distributing data into multiple visualizations. For example, multiple data views are useful for making side-by-side comparisons, or for viewing an overview and a detailed view simultaneously [192]. Baldonado et al. [15] propose that the cognitive overhead of interpreting a single complex visualization can be reduced by dividing the same information into multiple simpler views that can be observed in parallel. Each set of multiple views may contain only a subset of components from the full data set, however analysts can form mental links by switching their attention among them. As a caveat, browsing information across multiple views may incur additional costs such as greater required display space, increased memory load and effort for context-switching [15]. However, visual analytic research indicates there are cases when the benefits outweigh the costs [191].

Interactivity

Although actionable choices can sometimes be presented with a well-timed summary (e.g. the efficiency of a particular thermostat setting), many analytic tasks require a human decision-making component. The importance of interaction has been strongly highlighted in the visual analytics community. For example, two extensive surveys on interactive information visualization, one by Heer and Shneiderman [89] and another by Yi et al. [212] describe how interactions such as item selection, exploration of different representations, data filtering, and navigating through various levels of abstraction are essential to sensemaking in visual analytics. Although personal information visualization occupies a smaller scope than intensive domain expert analytic tasks, Huang et al. [100] note that human input can help to overcome the limitations of using automated data-mining techniques to identify patterns. Further, these operations should be coordinated across multiple views. For instance, using a technique known as *brushing* [231] causes a selection made in one view to be reflected through visual feedback (e.g. highlights) across related items on all views. Likewise, navigation such as zooming or filtering that selects a subset of data in one view can be made to concurrently filter the subsets of other views. For example, an examination of personal finances (**Figure 3**b), can be assisted by several automated processes (e.g. sorting, filtering, finding sums), but ultimately requires a user 'in the loop' to understand the data and make decisions.

1.3 Research Objectives and Overview

The primary research objective of this thesis is as follows:

Investigate the primary design aspects of spatial interfaces that fulfill the major design requirements to support in-situ, analytic tasks on future wearable computing devices.

This primary objective follows from the list of requirements posed above, each of which builds toward the final goal, aimed at proposing SAI as a viable platform for everyday, in-situ, analytic tasks. The path from this objective to the goal begins by establishing a grounded basis in the literature of existing spatial interfaces and affirms the chosen platform on which to base the work. From there, we investigate in depth each of the five requirements to guide the design of the corresponding system components. The steps along this path (outlined in **Figure 6**) are as follows:

a) We conduct a thorough literature review of spatial interface designs that involve 2D analytic workspaces. From this review, we discover strategies employed by spatial interfaces to support analytic tasks. To guide the following steps, we develop a design framework for categorizing these designs and aiding exploration of new designs.

b) From a variety of potential technologies, we make an informed choice of see-through HWDs as a suitable basis for our design exploration of SAIs. We develop a few scenarios depicting how this platform can support in-situ analytic tasks.

c) We explore an interface that supports the **mobility** and **multiple views** requirements for SAIs, and investigate the potential of this spatial interface for **interactivity**. Within the important *perspective* dimension of our design framework (see Chapter 2), we explore the design of a virtual window manager using an *egocentric* (body-centric) arrangement of information visualizations. Through a series of user studies, we explore the ideal arrangement of windows and confirm that switching between these windows using spatial navigation (head motion) is effective despite the field-of-view (FoV) limitations of a HWD.

d) Next we explore **integration** of displays into built environments. Balancing the *perspective* dimension against the previous step, we explore *exocentric* (world-fixed) layouts of virtual windows, embedded into surfaces in the surrounding environment. Our layout manager avoids occluding important objects with the integrated windows. In further support of the **mobility** requirement, the layout manager handles transitions between body-fixed and world-fixed versions of window layouts; relative positions of windows are kept as consistent as possible to preserve layout familiarity across different environments.

e) Within the window layouts described above, we explore **interactivity** on two important levels: 1) for manipulation of the window layouts to best support analytic

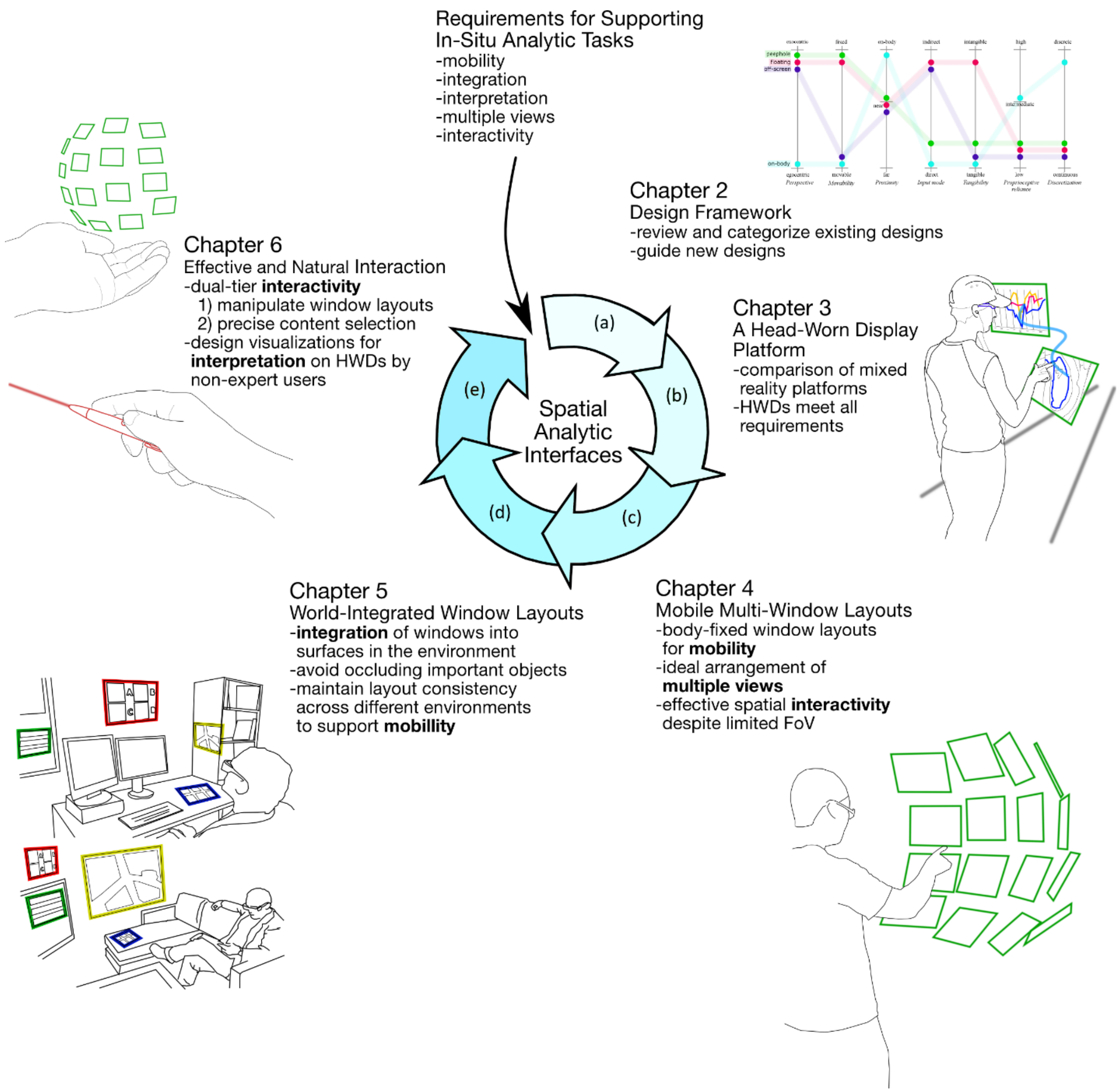

Figure 6. Thesis Overview

An overview of the research path followed in this dissertation, from the initial requirements to the final goal of proposing SAIs as a solution for in-situ, personal visual analytics. Each step along this path leads into to the next, with the complete path encompassing all of the initial requirements in a holistic exploration of SAIs.

tasks, and 2) for precise interaction with the visual analytic contents of windows. We design several novel interactive methods and interaction techniques, guided by dimensions of our framework. A secondary goal of this and the above sections is to create interface designs that support **interpretation** of the visual contents on the HWD platform for non-expert users.

1.4 Contributions

This thesis makes several contributions to the research community, as follows:

1) Introduce and define the concept of SAIs. SAIs leverage the advantages of spatial user interfaces, such as their support for proprioception and spatial memory, and apply them to the in-situ engagement in visual analytic tasks. We outline a set of requirements for such interfaces to improve over current technologies for in-situ analytic tasks.
2) Introduce a design space for SAIs based on a systematic literature review of designs that employ variations of 2D analytic workspaces in 3D spatial user interfaces. We encapsulate the recurring themes of these works in seven design dimensions and further categorize the designs into five common patterns.
3) Compare several platform options and describe how HWDs are well-suited to the requirements for in-situ visual analytics. We explore potential designs for this platform through several scenarios. We further develop the SAI concepts presented in these scenarios through a number of user studies, design explorations and implementations of several example interfaces on a HWD platform.

4) Explore in detail a breadth of human factors related to efficient information access on HWDs with a limited FoV. This investigation informs the design of a body-fixed, multi-window layout. We show empirically that this design is more efficient for switching views in an analytic task than two baseline view-fixed interfaces, based on currently available interface techniques.
5) Design an interface capable of integrating a multi-window layout into surfaces in the surrounding environment while minimizing interference with important objects in the scene. We further design this interface to keep the relative layouts consistent across environments to preserve user familiarity.
6) Design an interface that demonstrates effective dual-tier interaction for visual analytic tasks. We construct a prototype implementation that combines direct input from a hand-tracking camera and indirect input from a wearable ring device. Using this prototype, we demonstrate how this combination of input methods can be used to support a variety of interaction techniques for controlling both window layouts and their contained content, with seamless integration of these two layers in a single interface.

2 Design Framework

We propose that a new generation of spatial user interfaces can potentially overcome many of the limitations of current mobile technologies to provide suitable platforms for in-situ, analytic tasks. To learn the full range of benefits offered by spatial user interfaces, we conduct a systematic literature review of designs that use spatial interfaces. Although many spatial interfaces by nature use the three available dimensions of space, we specifically examine designs that use planar (2D) interfaces, since these are suitable for the display of visualizations for analytic tasks; many of the common and familiar information visualizations we use are primarily in 2D, whereas 3D visualizations are generally recommended only for specific purposes, such as visualizing spatio-temporal data. We call this design framework *Ethereal Planes*. We choose this name because, in many of the designs we find in the literature, the 2D surfaces of a user interface need not be restricted to the confines of a physical display. In many cases, designers superimpose these onto existing surfaces, including objects in the built environment or even onto surfaces of the human body. In some cases, they are situated in free space. This versatility and detachment from the computational objects that produce them provide these interactive planes with an 'ethereal' quality.

Ethereal Planes employs the concept of information spaces [65] in assisting the design of advanced and productive interfaces. Information spaces support intuitive computing interaction by mapping information to real world space, allowing us to look beyond the boundaries of the computing device and perceive information where it belongs: in the surrounding environment.

From our systematic literature review, we encapsulate the recurring design themes into seven design dimensions: perspective, movability, proximity, input mode, tangibility, visibility and discretization. By analysing common design choices from existing implementations we identify five common design categorizations: peephole, floating, off-screen, on-body, and palette. Further, we discuss several analysis techniques (e.g. tweaking, combining) that can help inspire new designs.

2.1 Background

Our goal in defining Ethereal Planes is to support the design of user interfaces for emerging technologies. However, we look beyond the individual technical challenges of these novel technologies towards a framework to encourage the development of everyday user interfaces for everyday applications. We encourage new and useful designs by providing a unifying foundation for the description and categorization of tools needed for manipulating spatially-distributed information. In this section we introduce the concepts of design frameworks and mixed-reality technologies.

Design Frameworks

Design frameworks are conceptual tools created to help designers conceptualize the nuances of particular technologies and formalize the creative process. Design frameworks have an established history in interface design, and have shown their value in providing terminology to categorize ideas [170] and organize complex concepts into logical hierarchies [163]. Design frameworks often accompany either the introduction of a previously unexplored concept (e.g. Graspable User Interface [65]) or the exploration of existing work in a new light (e.g. Ambient Information Systems [167], Availability Sharing Systems [93], and Ephemeral User Interfaces [51]).

Several frameworks related to spatial and mixed reality interactions have previously been developed for immersive virtual environments. For example, Bowman and Hodges [31] describe a framework outlining techniques for virtual navigation. Poupyrev et al. [166] present a taxonomy of virtual object manipulation techniques. Mine et al. [144] introduce a framework to leverage proprioception to assist interaction with virtual objects. Also, a well-known survey by Hinckley et al. [94] discusses many general issues relevant to spatial user interaction. In contrast to these previous frameworks, Ethereal Planes specifically addresses interface design for 2D, mixed reality information spaces and draws from work developed for a wide variety of mixed reality platforms.

In creating Ethereal Planes we used techniques also applied to HWD interface design by Robinett [180] and similar to those formalized in Zwicky's General Morphological Analysis [207]. This method treats a set of defined taxonomical terms as a set of orthogonal *dimensions* in a geometric *design space*. The resulting theoretical matrix provides a structure for objective classification and comparison. The methodical filling-in of this structure helps to categorize existing concepts, differentiate ideas, and identify unexplored terrain. This thesis follows three basic steps in the development and usage of our design framework:

1. Review of existing designs to distil a set of characteristic dimensions
2. Categorization of existing designs among these dimensions to identify both gaps and common usages
3. Generation of new designs through an analytic process of combining and altering design choices

Along these steps, our Ethereal Planes framework fulfils several purposes: The distillation from existing literature of a set of general but widely-encompassing design dimensions provides a taxonomy for designers, researchers, teachers and students to express their creations. The dimensional organization also helps the understanding of existing designs by providing a means to categorize them; by contrasting and comparing these, designers gain insight into general patterns and identify gaps in the dimensional framework where designs do not yet exist. Designers can then use this information to assist with the creation of new designs, either by applying the strengths of existing patterns to the correct contexts or thorough experimentation, by altering one or more dimension and then imagining the resulting implications.

2.2 Ethereal Planes Framework

The foundation of our Ethereal Planes design framework is an organizational taxonomy for classifying designs that incorporate virtual 2D workspaces.

Research Method

The taxonomy was the product of an extensive review of literature related to information spaces, and spatial interaction. Within this body of work, we found a subset of designs that embody the concept of Ethereal Planes. We began with a thorough archive search for papers exploring spatial user interfaces that occupy real world space, extending or existing fully beyond the limits of a conventional display screen. We focused on designs involving planar information spaces thus excluded designs that do not explicitly discuss 2D workspaces, for example those that involve navigating 3D workspaces through a 2D display. We also excluded papers that do not introduce distinct differences from previous designs, for example the use of an existing design in a new context or focus on the technology for implementing a known design. As a starting point for our search, we manually sifted through the previous 5 years' proceedings of CHI, UIST, ISWC and VRST, in which we expected to find the most recent and novel works of interest. We also conducted a tree search of references and citations from each of the initial papers we identified and of seminal papers on spatial interaction frameworks (e.g. [31,94,144,166]), which led to the inclusion of a variety of works from other sources. The final list, containing 34 papers, is not intended to be exhaustive, however represents a diverse selection of designs from which we draw. A complete list of all 34 designs in our survey, along with their dimensional classifications, may be found in **Table 1**.

From the papers in our literature review, we distilled a set of design dimensions using a bottom up approach resembling open coding. We began with 18 candidate dimensions that fit the concepts found in the reviewed literature, then iteratively reduced these into a set small enough to manage in a concise framework, yet containing enough dimensions to provide utility. We eliminated dimensions, for example, that expressed concepts that we deemed relatively insubstantial (e.g. fidelity), that were later incorporated into other dimensions (e.g. spatial reference frame) or that were substantial enough that treatment in our current framework would be superficial (e.g. co-located collaboration). Several important concepts that deserve further consideration are listed in a later section (Framework Extensions). This process resulted in seven design dimensions, listed in **Table 2**. We further organized the dimensions into three groups based on the strongest dependencies between them. This grouping is used to organize several resulting design recommendations.

Design Space Dimensions

Perspective denotes the conceptual viewpoint of the observer. To delineate this dimension, we borrow the terminology of *egocentric* and *exocentric* reference frames, used in early virtual reality (VR) literature and later included in a taxonomy for virtual object manipulation by Poupyrev et al. [166]. With the exocentric perspective, the viewer is an outside observer, whereas the egocentric perspective is immersive. These terms correspond to the sub-divisions of world- and body-based coordinate systems used in other taxonomies, such as that of Cockburn et al. [45]. Feiner et al. [63] expanded these to three possible reference frames for virtual windows, view-fixed, surround-fixed or object-fixed. Billinghurst [22] similarly refers to head-, body- or world-stabilized information displays. Hinckley et al. [94] use the terms relative and absolute gesture to denote motions in body-and world-centric space, respectively. In our framework, *egocentric* reference frames denote 'first person' (body-centric) reference points, such as the head or body, whereas *exocentric* frames are set relative to any object or other real-world (world-fixed) reference point.

Movability denotes whether workspaces are *movable* or *fixed* with respect to a given frame of reference. Fixed workspaces are indefinitely locked in place to their respective coordinate systems. Movable ones can be relocated in relation to their egocentric or exocentric reference point. In most contexts, we consider a hand-fixed information space as *movable* because it can be moved to different coordinate points within the reference fame, whether body- or world-centric. A mobile device display, for example, can be often relocated with respect to the user's head or body, thus does not usually qualify as *fixed*.

Proximity describes the distance relationship between an information space and its user. We use a set of regions drawn from neuropsychology [52,98] also used by Chen et al. [42]: *on-body* (coincides with pericutaneous space, on the body surface), *near* (peripersonal space, within arm's reach) and *far* (extrapersonal space, beyond arm's reach). The majority of implementations we examined involve interaction within arm's reach, often by direct input (e.g. [36]) or with a handheld device (e.g. [226]). Some systems allow interaction with distant objects, particularly those for immersive virtual worlds or for outdoor use (e.g. Augmented Viewport [96]). Other researchers have explored the human body as an interface (e.g. [86]).

		Reference Frame				Spatial Manipulation							Spatial Composition				
		Perspective		Movability		Proximity			Input mode		Tangibility		Visibility			Discretization	
	Design	ego-centric	exo-centric	movable	fixed	far	near	on-body	direct	indirect	tangible	intangible	high	intermediate	low	continuous	discrete
Peephole	Touch Projector [27]		•		•	•	•		•		•		•			•	
Peephole	Dynam. Def. Info. Spaces [36]		•		•		•		•		•		•			•	
Peephole	mSpaces [38]		•		•		•		•		•		•			•	•
Peephole	Chameleon [66]		•		•		•		•		•		•			•	•
Peephole	Bonfire [105]		•		•		•		•		•		•			•	
Peephole	X-Large Virtual Workspaces [109]		•		•		•		•		•		•			•	
Peephole	Pass-Them-Around [128]		•		•		•		•		•		•			•	
Peephole	Augmented Surfaces [177]		•		•		•		•		•		•			•	
Peephole	PenLight [188]		•	•	•		•		•		•		•			•	
Peephole	MouseLight [189]		•	•	•		•		•		•		•			•	
Peephole	PlayAnywhere [225]		•		•		•		•		•		•			•	
Peephole	Lightspace [226]		•		•		•	•	•		•		•			•	•
Peephole	Peephole Displays [230]		•		•		•		•		•		•			•	
Floating	Friction Surfaces [6]		•		•		•			•		•	•			•	
Floating	Wearable Conference Space [22]		•		•		•			•		•	•			•	
Floating	Touching the Void [40]		•		•		•		•			•	•			•	
Floating	Hybrid int. in VEs [82]		•		•		•			•		•	•			•	
Floating	Windows on the World [63]	•	•	•	•		•			•		•	•			•	
Floating	Augmented Viewport [96]		•		•	•				•		•	•			•	
Off-Screen	Portico [12]		•	•			•			•	•		•			•	
Off-Screen	SideSight [34]		•	•			•			•	•		•			•	
Off-Screen	Off-screen Pan and Zoom [104]		•	•			•			•		•	•			•	
On-body	Chen et al. [42]	•	•	•	•			•	•		•			•		•	•
On-body	Imaginary Phone [78]	•		•				•	•		•			•			•
On-body	OmniTouch [84]	•		•			•	•	•		•			•		•	•
On-body	Skinput [86]	•		•				•	•		•			•			•
Palette	de Haan et al. [83]	•		•			•		•		•		•			•	
Palette	Lindeman et al. [124]	•		•			•		•		•		•			•	
Palette	Transparent Props [183]	•		•			•		•		•		•			•	
Palette	Personal Interaction Panel [199]	•		•			•		•		•		•			•	
	Virtual Shelves [121]	•			•	•				•		•			•		•
	Imaginary Interfaces [77]	•		•			•		•			•			•	•	
	AD-Binning [87]	•		•			•		•		•	•			•		•
	Piles Across Space [221]	•		•			•			•		•			•		•

Table 1. Ethereal Planes Survey Overview
We included 34 designs in our survey for Ethereal Planes. From these we parsed seven design dimensions, organized into three related groups. Projecting the dimensions back onto the designs reveals five categories. Designs that do not cleanly fit their defined categories are highlighted in yellow. Some additional example designs that do not cleanly fit a category are shown in the bottom four rows.

Input mode falls coarsely into two camps, *indirect* and *direct*. Indirect input includes cursors, ray-casting and variations of these methods. Direct input includes input using direct touch by hand, fingertip or stylus as well as virtual 'touch' with intangible surfaces (e.g. [40,78]).

Tangibility defines whether an information space is mapped to a surface that can be touched. Our frame work classifies implementations as either *tangible* or *intangible*. *Tangible* interfaces often leverage surfaces in the nearby environment, such as a wall (e.g. [36]) or device screen (e.g. [226]) and benefit from haptic feedback. *Intangible* designs typically make use of 'in-air' gestures (e.g. [77]) for user input.

Group	Dimension	Values		
Reference Frame	*Perspective*	egocentric	exocentric	
	Movability	movable	fixed	
Spatial Manipulation	*Proximity*	far	near	on-body
	Input mode	direct	indirect	
	Tangibility	tangible	intangible	
Spatial Composition	*Visibility*	high	intermediate	low
	Discretization	continuous	discrete	

Table 2. Ethereal Planes Design Framework Dimensions
From our literature review, we define a design framework with seven dimensions, organized into three groups defined by inter-dependencies of the related dimensions. We assign each dimension a discrete set of values, although in some cases the range of potential values may be continuous in practice.

Visibility describes the amount of visual representation available in an interface and also determines the degree to which spatial memory relies upon proprioception. Our framework uses three levels of visibility, *high*, *intermediate* and *low*. *High* visibility means that the information space is largely or fully visible. *Intermediate* visibility means some type of viewing constraint is present, for instance if only a small section of the workspace may be seen at one time (e.g. [230]). *Low* visibility implies that information management relies very little or not at all on visual feedback (e.g. [77]).

Discretization specifies whether an information space is *continuous* or composed of *discrete* units. The majority of designs in our survey use *continuous* space. Examples of *discrete* mappings are the body-centric browser tab mappings described by Chen et al. [42] and the bins Wang et al. [221] placed around a mobile device for sorting photos.

Dimensional Interdependencies

While the dimensions of a design space are ideally orthogonal, dependencies between dimensions are rarely entirely absent. As a case in point, some choices in the Ethereal Planes dimensions will have implications for others. We clustered the dimensions by their closest dependencies into groups we call *Reference Frame*, *Spatial Manipulation* and *Spatial Composition* (**Table 2**). Here we discuss some of the trade-offs between design choices within each of the three groups.

Reference Frame – Perspective and *movability* together encompass the concept of a spatial reference frame. Combinations of these two dimensions are summarized in **Figure 7**. Different reference frames are better suitable for different types of applications.

In a mobile scenario, an *egocentric* perspective is more useful, since it will move along with a user on-the-go. In collaborative scenarios, *exocentric* space is more appropriate, since users will benefit from a shared, world-based reference frame, as is the case with a real-world, wall-fixed whiteboard. *Exocentric* frames are also useful for situating information spaces in the contexts where they are most practical [65]. However, in free space interactions, Hinckley et al. [94] note that *egocentric* coordinate systems are easier for users to comprehend and manipulate than *exocentric* frames.

Fixed information spaces are useful in situations where spatial memorability is important, for example in the placement of application shortcuts [121]. Once learned, objects in *fixed* spaces can also be recalled with the aid of proprioception [78,121,230]. *Movable* workspaces, conversely, are better for short-term memorability such as when the information contents are short-term, volatile or highly dynamic.

Spatial Manipulation – The three dimensions of *proximity*, *input mode* and *tangibility* are related to the manipulation of information spaces and of data and objects within them. **Table 3** provides examples of relevant combinations between these dimensions. For various reasons, some combinations have no existing counterparts in our Ethereal Planes-related literature. With indirect input, for example, the concept of *tangibility* becomes less relevant, thus we do not include *tangibility* under the indirect column of the table. Conversely, it is difficult to imagine direct input with *far* proximity, thus no examples appear in our survey (although this does not mean that some conception of such a concept cannot be realized in future).

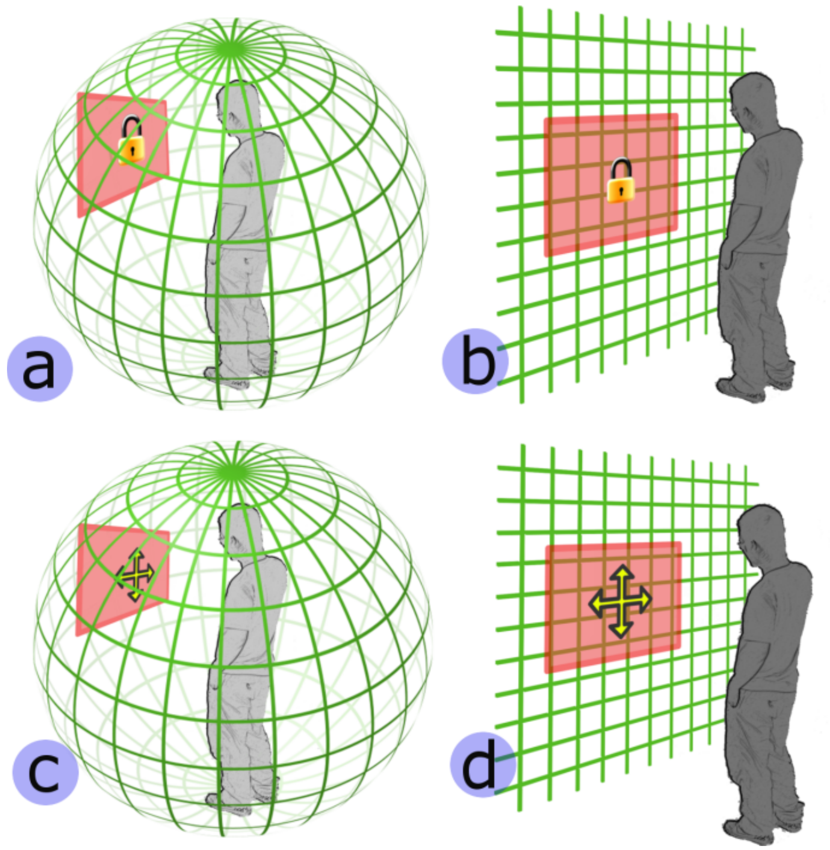

Figure 7. Reference Frame Breakdown
The first two dimensions, perspective and movability, compose the first group of the Ethereal Planes design space, *Reference Frame*. The resulting four potential combinations are depicted by diagrams in this matrix: (a) fixed-egocentric, (b) fixed-exocentric, (c) movable-egocentric and (d) movable-exocentric. Egocentric reference frames (a, c) are body-centric, while exocentric frames (b, d) are world-fixed. Content can be either fixed (a, b) or movable (c, d) within these frames.

Input mode is dependent on *proximity*: whereas *indirect* input allows interaction with surfaces that are beyond reach, *direct* input is intuitive when the interface lies within reach. *Direct* input is practical with *on-body* surfaces since it leverages proprioception. Leveraging available surfaces, whether body or other, also assists motor precision [124].

	Input mode	*direct*		*indirect*
	Tangibility	*tangible*	*intangible*	
Proximity	*on-body*	Skinput [86], OmniTouch [84]		
Proximity	*near*	Peephole displays [230], Cao et al. [36]	Touching the void [40], Imaginary interfaces [77]	Sidesight [34], Windows on the world [63]
Proximity	*far*			Virtual shelves [121], Augmented viewports [96]

Table 3. Spatial Manipulation Breakdown
Three design dimensions, proximity, input mode and tangibility, compose the group *Spatial Manipulation*. This table shows several examples of how different designs fit into the group. Greyed-out cells represent areas of the design space where no matching examples were found in our literature survey.

Tangibility is influenced by the technology platform chosen. Projection-based interfaces are often *tangible*, since a projection surface is required. Stereoscopic displays (i.e. Caves, some HWDs) often use *intangible*, virtual surfaces, although information spaces are sometimes intentionally set to coincide with physical surfaces [199]. In free space, researchers have found that *indirect* input is faster, less fatiguing and more stable [6,94,205] than *direct* input. However, *direct* input is intuitive and can make use of expressive gestures, thus may be desirable even without the aid of a *tangible* surface. Our survey turned up many designs using *direct* input both with (e.g. [36,84]), and without (e.g. [40,77]) *tangible* surface contact.

		Discretization	
		continuous	*discrete*
Visibility	*low*	Imaginary interfaces [77]	Virtual shelves [121], Piles across space [221], mSpaces [38], body-centric browser tabs [42]
	intermediate	Peephole displays [230]	Skinput [86], Chameleon [66]
	high	Pen light [188], Mouse light [189]	

Table 4. Spatial Composition Breakdown
The remaining pair of dimensions, visibility and discretization, compose the final group, *Spatial Composition*. Several example designs are located in their respective cells of the table. Since discrete spatial mappings are often used to support spatial memory or proprioception in interfaces with little visual feedback, no designs with a combination of high visibility and discrete mappings were found in our survey.

Spatial Composition – Together, *visibility* and *discretization* contribute to the way information is organized spatially. One important factor related to these dimensions is spatial memory. Spatial memory is important in many of the interface designs considered in our survey, particularly when the information spaces are not confined within the boundaries of a typical display screen (e.g. [230]). **Table 4** shows examples of different pairings between *visibility* and *discretization*. The majority of interfaces represent information visually; however, some present little or no visual information. Spatial memory can be built either purely visually, or by muscle memory, although many designs leverage some combination of both (e.g. [84,230]). Designs with little or no visual feedback are more likely to rely highly on proprioception for object recall (e.g. [77,121]). *Discrete* spatial mappings are commonly used with interfaces with *intermediate* or *low* visibility. When little or none of the interface can be seen, designers can instead leverage spatial memory or proprioception, (e.g. Virtual Shelves [121]). In such cases, *discretization* is often leveraged to make recall manageable.

2.3 Framework Applications

We created our Ethereal Planes framework to guide our own research and also to assist future designers. Here we discuss how our framework can be used to categorize and compare existing designs as well as aid the creation of new designs.

Categorizing Existing Designs

A fundamental aspect of any framework is its descriptive capacity. The ability to clearly describe aspects of a design allows it to be deconstructed according to its various facets of functionality. Functional decomposition provides a formal structure to allow designs to be compared or contrasted in a methodological manner. Comparing designs may be useful for pedagogical purposes, or for designers reflecting on a number of previous solutions that may potentially be applied to a current problem. A framework's descriptive capacity also allows designs to be stored into repositories and later searched to support various formal techniques for concept generation [26].

To show how Ethereal Planes can be used to describe existing designs, we apply it to the works from our literature review. For each design, we assigned dimensional values and classified the results, which provides us with a methodical system to contrast and compare these different designs. We acknowledge that our framework does not provide an absolute partitioning in which designs fit cleanly into the dimensional values. Rather there are many cases where different values apply to multiple presented concepts or the chosen values are open to interpretation. However, the goal of our framework is not to provide a set of arbitrary sorting bins, but to make the designer aware of important design choices and help them weigh the potential benefits of these choices.

Several distinct categories of similar designs emerged from our analysis, each of which we describe in detail below. Although these five categories represent only a small geometric region of the full design space, we found that the majority of reviewed designs (30 of 34) are a very good fit to one of them. As with the assignment of dimensional values, these categories are not absolute, thus we include minor variations that fit closely to the overall character of the group. A few more diverse exceptions are discussed in the following section.

Peephole – In the first and largest of our categories, we group concepts that build on the *spotlight* and *peephole* metaphors. These designs allow interaction through 'peephole windows' that are moved around the surface of a 2D workspace. Both are conceptually similar with their main difference being the technology used: Whereas *peephole* interaction implies the use of spatially aware mobile devices, the *spotlight* metaphor typically refers to the use of mobile projectors. The common moniker of 'peephole' interaction was coined by Yee [230], but is a direct descendant of Fitzmaurice's Chameleon. The common theme motivating these designs is to expand the workspace beyond the limited boundaries. To prevent getting lost in a large, mostly invisible space, the workspace remains world-fixed while the device user navigates the content within. Whereas the original Chameleon [66] implementation used the *discretized* space of a spreadsheet application, most variations use *continuous* 2D space. Several other variations, not discussed here, explore 2D 'image-plane' representations of 3D space. Variations from our research include: Touch Projector [27], mSpaces [38], Chameleon [66], Pass-them-around [128], Peephole displays [230], dynamically defined information spaces [36], PenLight [188], Mouse-Light [189], Augmented Surfaces [177], PlayAywhere [225], Lightspace [226], Bonfire [105] and X-Large virtual workspaces [109].

Floating – This group contains various instantiations of virtual windows that appear to *float* in mid-air. A common goal of these designers is to import the familiar characteristics of ubiquitous 2D applications into an immersive environment. *Floating* windows have often been used to implement auxiliary input controls such as panels, dialog boxes and menus, in immersive VR environments [82]. Since mid-air displays are *intangible*, designers often use *indirect* input modes such as mice [63,96] or ray-casters [6]. Chan et al. [40]provide an interesting exploration of *direct* interaction with *intangible* displays. Other variations include: Windows on the World [63], Wearable Conferencing Space [22], Friction Surfaces [6] and Augmented Viewport [96]. Most of these use *exocentric* information spaces, however some HWD implementations [22,63] provide the option of *egocentric* floating windows for mobile users.

Off-Screen – This category includes designs that allow *indirect* input in the 'off-screen' region that surrounds a device's periphery. As in the peephole concept, *off-screen* designers address the problem of limited screen space by extending the theoretical plane of a device's screen into surrounding space. However, these systems are easily portable, allowing the surrounding workspace to be conveniently repositioned. They also avoid occlusion with *indirect* input, and are useful for navigational operations such as panning and zooming. We generalize this category as *exocentric* because two of the included designs (SideSight [34] and Portico [12]) use a device placed on a surface. However, the third example (off-screen pan and zoom [104]) is *egocentric*, since it uses a handheld device.

On-body – Another convenient tangible surface is the human *body*, used by the designs in this category. In many instances, a hand or arm doubles as a convenient projection surface in lieu of a wall or table, and is a convenient, always-available place to store buttons or task shortcuts. Body parts have the primary benefit of assisting target acquisition with proprioception, as evidenced in Harrison et al.'s Skinput [86]. Variations on this theme include Imaginary Phone [78], OmniTouch [84] and Chen et al.'s Body-centric prototype [42].

Palette – These designs align the information space with a handheld *palette*, such a paddle or transparent sheet. This use of a handheld plane allows bimanual interaction, which can facilitate task performance [124]. Handheld tangible surfaces have commonly been used in immersive environments, since *tangible* surfaces provide increased speed and control over *intangible* floating surfaces [124]. Variations include the Personal Interaction Panel [199] and various similar implementations [83,124,183].

In **Figure 8** we provide a visual summary of the major design categories in a parallel coordinates graph. This graph shows the values of each category along the seven design dimensions. This figure fulfils several purposes: 1) It, shows where designs are similar and where they differ, which allows them to be easily compared and contrasted. 2) It shows clustering within the dimensions, including commonly occurring values (e.g. *near proximity* or *high visibility*) and commonly joined pairs (e.g. *exocentric-mixed* or *direct-tangible*), which reveals similarities between apparently different designs and highlights common approaches. 3) In contrast, it exposes the areas of the design space that are under-utilized (e.g. *far proximity* or *intangible*), which allows gaps to be explored, as discussed in the following section.

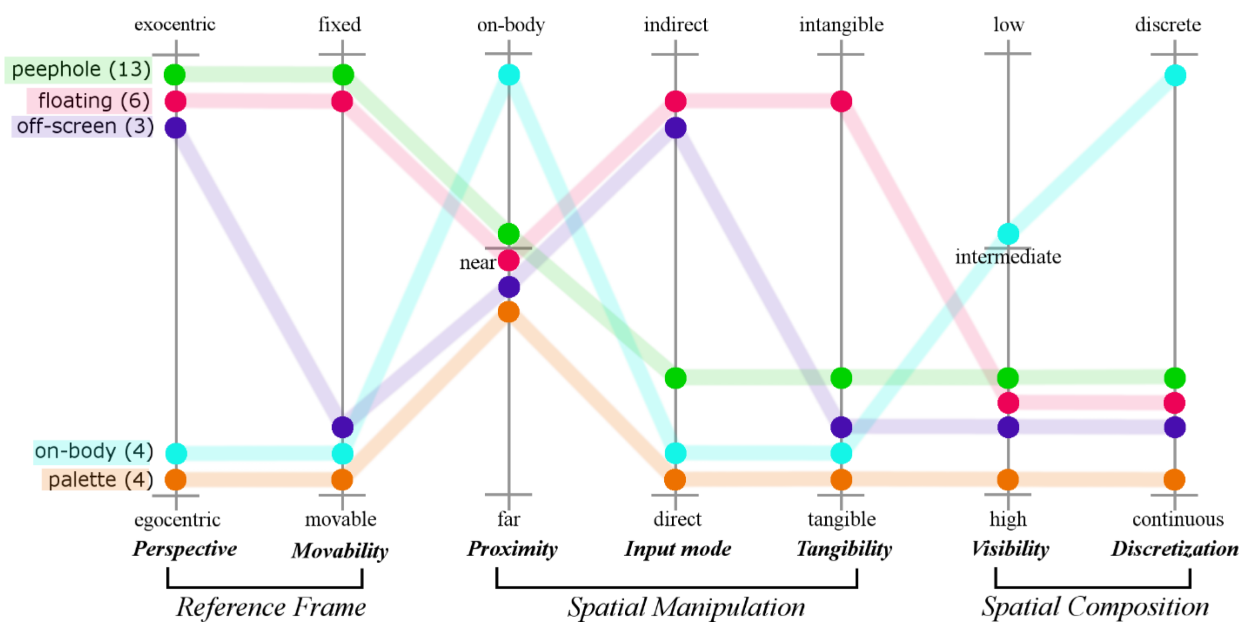

Figure 8. Ethereal Planes Design Categories
A parallel coordinates graph shows the main design categories found in our analysis, as they appear mapped across the seven dimensions of our design space. This visualization allows the design categories to be easily compared, and shows areas of the design space that contain relatively many or few designs.

Filling Gaps, Tweaking and Combining

Beyond classification and comparison of existing designs, one purpose of a framework is to inspire and guide new creations. To show the generative potential of Ethereal Planes, we discuss several analytic processes that can be undertaken with our framework. Following Zwicky's morphological analysis [179], we explore three primary operations that can be used to transform our prior set classifications into ideas for new designs, by identifying *gaps* in the matrix, by '*tweaking*' (altering) existing designs, or by *combining* two or more of them.

The first way to think about new designs is *filling gaps*; to look for valid combinations that have not been tried. Zwicky, who championed morphological analysis, viewed such gaps as opportunities to inspire creativity. By applying a morphology to our framework, its dimensions can be viewed as a seven-dimensional matrix, where each cell is a different combination of chosen values. Theoretically, this matrix has 288 unique design patterns. This number seems remarkable, considering that we were able to classify a large number of designs into only a handful of patterns. What then is the explanation for this difference? One primary reason is the number of interdependencies between the framework dimensions. Because the dimensions are not purely orthogonal, many of the possible combinations may be considered invalid. For instance, *direct* input with *far* information spaces seems impractical. However, the Ethereal Planes design space is still relatively unexplored and perceived dependencies may in fact be a result of attachment to prior paradigms. For instance, the most common reference frame types in the explored literature are *exocentric-fixed* and *egocentric-movable*, which correspond respectively to the most common types of real-world displays: desktop monitors and mobile devices. As designers gain more experience with mixed reality applications, some of the combinations that appear invalid may be explored with new and unconventional concepts. For example, the *direct-far* combination mentioned above may be solved by introducing a mechanism for controlling stretchable virtual limbs. On the other hand, *indirect-on-body* interaction might be found useful when looking at one's self in a mirror. In this manner, the Ethereal Planes framework is useful for plotting existing designs across the design dimensions, providing a formal tool to help designers to identify new ground and inspire unique creations.

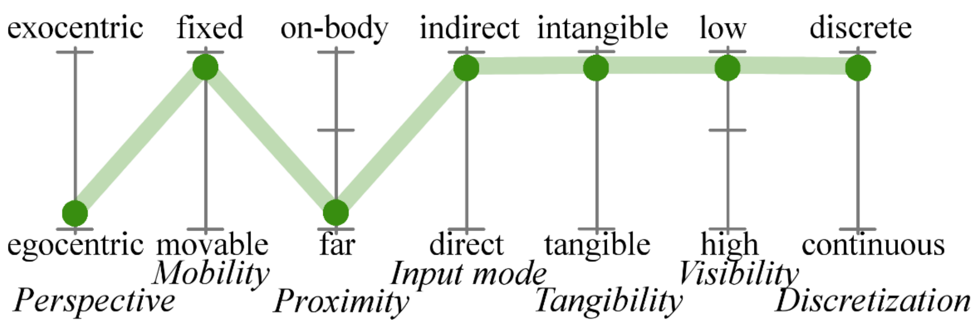

Figure 9. Example Design #1: Filling Gaps
The Virtual Shelves design of Li et al. [121] **is an example of a design that defies easy categorization, as it holds a unique position in the design space. The parallel coordinates graph makes it easy to see adjacent value pairs that are distinct from any of the designs in the five main categories shown in Figure 8**.

One example of a design that falls between the gaps of the categories we identified is the Virtual Shelves implementation described by Li et al. [121]. With the Virtual Shelves interface, selectable objects, such as icons, are distributed in an *egocentric* sphere around the user. The user relies on spatial memory to make selections using a ray-casting metaphor, thus the objects are conceptually at a *far proximity*. This design combines some dimensional values not found in any of the main categories (**Figure 9**), such as an *egocentric-fixed* reference frame and *low visibility* with *discrete* space. The parallel coordinates visualization makes it easy to see that this design creates a unique pattern in the Ethereal Planes design space.

A second method for creating new designs is *tweaking*; rather than create a new combination from scratch, we can change one or two dimensions of existing patterns and imagine the resulting implications. In fact, one such example we identified in our literature review is the Imaginary Interfaces design of Gustafson et al. [77]. It is similar in nature to the *palette* category, however the user can 'draw' objects such as letters or mathematical functions with their fingertip on an intangible and invisible surface. This unusual design breaks the conventions of previous patterns by combining *low* visibility with a *continuous* workspace (**Figure 10**). Although only two dimensions are changed, the result introduces some significant design challenges, many of which are addressed in this novel work.

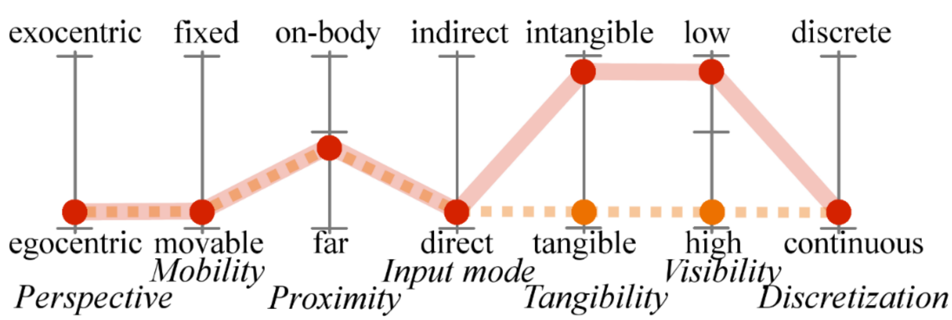

Figure 10. Example Design #2: Tweaking Existing Designs
The Imaginary Interfaces design of Gustafson et al. [77] **is an example of how new design can be created by *tweaking* existing ones. This design (solid path) varies from the *palette* category (dashed path) only in the *tangibility* and *visibility* dimensions.**

One other way to generate new ideas is to *combine* two or more existing patterns. An example of this type was also identified in our reviewed designs, in the AD-Binning implementation of Hasan et al. [87]. This interface extends the interaction plane of a mobile device screen into space around the device for making *discrete* item selections. This design has many dimensional values in common with palette category (*egocentric*, *movable*, *near proximity*, *direct input*), but also some in common with Virtual Shelves (*intangible, invisible, discrete* space). Combining these dimensions creates a new hybrid pattern, as seen in **Figure 11**. A similar fit to the framework was found in the Piles Across Space implementation of Wang et al. [221], which was designed for sorting photos into virtual piles around a desktop monitor. Designers of future interfaces can benefit from a design space that provides a conceptual workspace for trying new combinations.

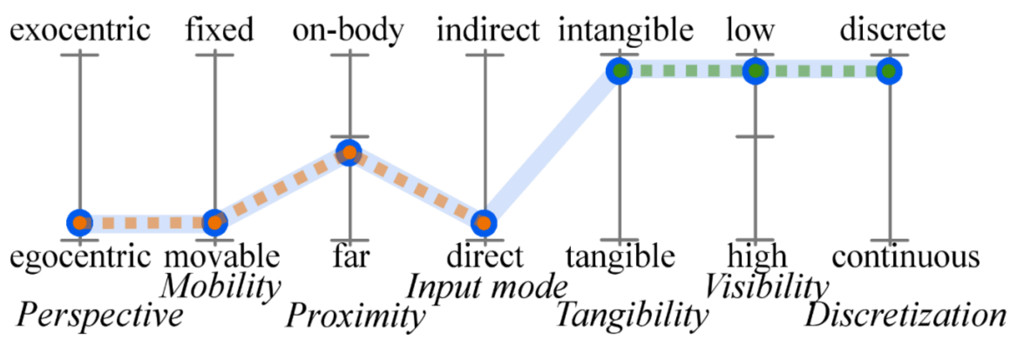

Figure 11. Example Design #3: Combining Existing Designs
The AD-Binning design of Hasan et al. [87] **shows how new designs can be created by *combining* others. This design (solid path) shares some dimensional values with the *palette* category (orange) and others with the Virtual Shelves design (green).**

One particular instance where combining existing designs can be useful is to support multiple interface 'modes' within a compound design. For example, imagine a sketching application with read and write modes. Suppose a series of sketches are distributed in an egocentric sphere, floating around the user, which can be viewed using a mobile screen. When editing the sketches in write mode, the user uses the display as a *peephole*, since it provides a *tangible* surface to assist drawing in *continuous* space. To make drawing easier, the sketches are mapped to a single stationary (*exocentric*) plane, so the user doesn't need to change the device orientation. When viewing the sketches in read mode, however, the user can simply hold the device in one place and use her second hand as a pointer; the user knows the *discrete* location of each sketch in the *egocentric* sphere and whichever one she points to appears on the display. A single dimension can also act as a 'mode switch' within a single design. Imagine for instance an image browsing application. The user can have both a collaborative mode and a personal mode. To support sharing, the collaborative mode uses *exocentric* space, whereas the personal mode is placed in *egocentric* space.

2.4 Framework Extensions

We acknowledge that there are limitations to our Ethereal Planes framework which may make it seem incomplete in certain contexts. However, we view Ethereal Planes as a core template that can be modified to suit a designer's needs, rather than a final product that fits all circumstances. Here we briefly discuss several potential extensions of our framework. These extensions include ideas that we initially attempted to introduce into our list of framework dimensions, but warrant deeper consideration at a higher level than is possible with the initial framework we introduce in this paper. Each of these topics requires several dimensions of its own that could constitute a separate layer of a more complete framework. In each case, these dimensions must be drawn from an additional body of literature and must be considered at a higher level than the basic interaction concepts of our initial framework.

Multi-modal interaction: Our input dimension takes into account only the paradigms of pointer selection and direct manipulation. This dimension could be expanded to include other input modes, particularly voice. The *visibility* dimension could similarly be expanded to consider non-visual output modes such as audio output. Such extensions would allow our framework to be extended to the design of interfaces for people with motor-skills or visual disabilities.

Co-located Collaboration: One of the applications of our framework is for collaborative scenarios. HWDs connected by network can be configured to allow multiple people to view the same virtual workspace from different perspectives [2]. Our framework could be extended by taking into consideration the large body of research on multi-surface environments. The modified framework should include aspects pertaining to the movement of content between surfaces and consideration of public vs private content [75].

Beyond 2D Surfaces: Our current framework focuses on 2D surfaces, although it could be extended to handle 3D objects. Such an extension should include additional dimensions to handle manipulation and viewing (grasping, rotation) of 3D objects. It should also include dimensions that take into account occlusion caused by the object's relative orientation or clutter from multiple objects.

3 A Head-Worn Display Platform

In the preceding chapter, we explored a wide variety of technologies that could potentially be used for a platform to support SAIs. For the scope of this thesis, we focus on one particular technology, see-through HWDs. In this chapter, we summarize a number of available technologies and compare their benefits and drawbacks. From this comparison, we single out HWDs, and describe in detail how the features of this class of device can potentially support all the requirements for in-situ analytics. This discussion provides a basis for the remaining chapters of this thesis, which explore various elements of SAI on a HWD platform.

3.1 Mixed Reality Platforms

The phrase 'mixed reality' is an overarching term that describes the combination of real and virtual objects [142,176]. This term encompasses augmented reality (AR), where virtual objects are superimposed on the real world, and VR, which immerses the user in a virtual world. Buxton and Fitzmaurice [35] identified three potential mixed reality platforms for that can be used to support virtual information spaces: HWDs, Caves, and handheld devices. More recently, projection has become commonly used in a format known as spatial augmented reality (SAR). These technologies have all advanced significantly to become staples of mixed reality. Each has its advantages and limitations, which we discuss below.

HWDs

Mixed reality has its roots in the see-through HWD technology introduced by Sutherland [197] in the 1960s, who envisioned the 'ultimate display' [196] that can control physical matter. A wide variety of realizations have undergone development since. Many advances in 3D interface design have occurred as a result of VR research since the early 1990s. VR has seen a recent resurgence in popular culture as advances in hardware have progressed to the stage where relatively light-weight, low-latency devices such as Oculus Rift [156] and HTC Vive [99] are entering the market.

Optical see-through HWDs show content on transparent lenses, and are most widely known through the introduction of Google Glass [73], which revealed deeply held user concerns about privacy and social acceptability. In contrast to Glass, which was designed for micro-interactions on a single (monocular), small, display placed in the user's peripheral view, another class of see-through HWDs places dual, binocular displays directly in the user's line of sight. Such devices allow objects to be superimposed stereoscopically in 3D space, and are ideally suited for the development of SAIs. One advantage of the wearable form factor of HWDs is that they provide a convenient location to place sensors for tracking the user's hands or the external environment. Microsoft's HoloLens [139], for example, can construct a model of the user's surroundings in real time and use this information to integrate virtual displays on nearby walls. As miniaturization continues, we may soon expect devices that look similar to typical eyewear, which will help reduce the barriers to social acceptance of HWDs for everyday use.

Perhaps the strongest potential advantage of HWDs is their potential for collaboration. Multiple devices can be networked to reveal the same virtual content to multiple users from each person's perspective.

CAVEs

CAVE technology was initially developed by a group of researchers at the Electronic Visualization Laboratory at the University of Illinois at Chicago, and was intended to overcome the limited viewing field of HWDs to provide a tool for scientific visualization [47,48]. CAVEs immerse their users within a volume surrounded by projection or other display surfaces to provide a display that covers the entire human FoV. However, various realizations of this technology may vary from a single large display wall to a nearly-full enclosure including projections on the floor and ceiling. Like HWDs, CAVEs are capable of stereoscopic display of 3D content, however users are restricted to the physical boundary of the display hardware. While CAVEs face limitations in the size and expense of the necessary hardware, there are no physical constraints on the amount of computational power used to drive this hardware. As a result, such systems are often capable of low-latency and high-precision (sub-millimetre) motion tracking. Although techniques such as shutter-glasses (e.g. [2,198]) allow multiple users, such techniques scale to only a small number of users within a shared space.

Handheld Augmented Reality (HAR)

As portable display technologies such as tablet computers and smart phones became widespread in the first decade of the 21st century, researchers began developing systems to leverage these devices as AR displays. With handheld augmented reality (HAR), a device display acts as a 'window', through which users view virtual content overlaid on a backdrop provided by the device's embedded camera. Early versions of this concept relied on external tracking systems and servers to create real-time effects [72,153]; however, as handheld devices increased in computational power, researchers began developing fully self-contained implementations [23,218]. Applications that incorporate AR are now commonplace on smartphones. These require tracking the device locally, commonly done with the assistance of external fiducial markers [108]. Toolkits such as Vuforia [217] now allow users to define their own fiducial markers based on arbitrary images, and even support tracking relative to predefined 3D objects. Tracking of rotation is also possible using the phone's internal IMU sensors [95]; however, position tracking with the IMU is not possible due to uncontrollable drift, unless combined with a secondary method to provide an external reference frame [233]. Although virtual content can only be viewed through the device's display, one advantage of touchscreen devices is that they support a haptic surface for interaction through gestures on the device's screen. Several techniques have been developed for interacting with virtual content in this manner [79,117,149].

Spatial Augmented Reality (SAR)

Spatial augmented reality (SAR), originally demonstrated by Raskar et al. [173–175], uses projectors to overlay virtual content on real-world surfaces. This concept is similar to the use of transformed projection in CAVE displays; however, projections are mapped to the underlying structure of the given projection surface [177,209]. Unlike CAVE displays, projection mapping allows SAR to be used in everyday environments, but it comes with several drawbacks. First, it requires rooms to be purposefully furnished with projection equipment, although it is possible to create mobile SAR implementations through the use of handheld projectors [36,172]. Second, projecting in uncontrolled environments can cause unpredictable colour distortion due to the inherent hue and reflectivity of a given surface. Solving this requires either controlling the projection area's surface make-up, or compensating the projection image [103]. The recent availability of low-cost depth sensors such as Microsoft Kinect [140] have made SAR increasingly popular as it becomes easier to scan environments in real time. Further developments, such as the RoomAlive toolkit recently released by Wilson et al. [141] for calibrating multiple depth cameras, will increase the general availability of this technology.

3.2 Head-Worn Displays as a Platform for SAIs

Upcoming see-through HWD technologies provide many opportunities for meeting our set of five requirements for SAIs in that they are wearable, allow spatial-user interfaces, augment the user's surrounding, and can contain embedded sensors. We elaborate on these opportunities below, with a summary in **Table 5**.

Wearable

Being wearable devices, HWDs are inherently mobile and the interface is always available. This property makes them ideal devices for in-situ visual analytics. HWDs can be worn in virtually any situation – at home, during work, or while on the go – making them more versatile than projection-based approaches that require equipment to be installed. Also, unlike current mobile devices, they can provide information with hands-free access, making their use practical in situations when the user's hands are occupied, such as carrying groceries or holding on to a subway handrail.

Spatial User Interfaces

HWDs are capable of providing a far richer experience than is available with current mobile technology. Embedded sensors and stereoscopic viewing capabilities can provide an 'immersive' experience, where virtual objects can be made to appear in physical space, or integrated with surrounding real-world objects. Whereas the small display of a smartphone requires its user to divert their attention from the outside world to a handheld object, HWD content can instead be integrated with our surroundings. Thus HWDs have potential to attract our attention toward, rather than away from, objects in the real world. This level of integration provides a range of display possibilities, from ambient displays that require little attention, to a set of multiple display panels laid out in space.

Furthermore, any region of real world space can be used to host a virtual display, thus the amount of 'display space' available for use by HWDs is limited only by the ergonomic viewing constraints of the user. Multiple displays can be situated in space, for example in a ring or sphere that follows the user as she walks, or arranged to coincide with nearby surfaces such as walls or desktops (**Figure 3**b). Switching between different views laid out in space provides a more natural and efficient experience than navigating between application views on a display that is fixed in the user's line-of sight [56] or on a handheld mobile device. Since rearranging views does not require moving physical objects, displays can easily be placed on any existing surface or even in mid-air. Adding additional displays for multiple views does not require the expense of additional monitors, and can be used anywhere for in-situ analytic tasks.

Augmentation

Virtual displays can produce some effects that are not easily obtainable with conventional display technologies. For example, a display can easily change size, say to shrink out of the way when someone enters the room and interrupts the analytic task. Visual links can connect related items across different visualizations such as items that are jointly highlighted in a coordinated selection [46]. Such links have been shown to help users find related entities more quickly than highlights alone in a desktop environment [192]. On physical displays, visual links can only connect items across views within the same display space, whereas with virtual 'floating' displays, such links can connect views across interstitial space. While a similar effect is possible by rendering a 3D environment on a flat display [46], the spatial user interface of a HWD allows the user to actually move among and between the visualizations and links to gain the best perspective without the need for abstract virtual navigation. In this spatial environment, such links can serve the dual purpose of guiding users' attention to related items, while also guiding users to the physical locations of other displays distributed in the physical surroundings. Other possibilities with greater leverage on AR techniques can be imagined to integrate information more directly with the surroundings. For example, a building that contains a hotel or restaurant can be overlaid with information such as reviews, menus or room availability. Aggregated location tracking data can be overlaid on the floor of a plaza to show the paths of various visitors. Or in a home environment, different rooms can be overlaid with visualizations showing trends about temperature, overall energy consumption and human traffic flow, which could provide a useful context for programming a thermostat and control of air vents. Virtual displays can also be used in conjunction with physical displays, for example to provide peripheral display space for sorting bills around the screen of a home desktop computer, or to provide a large overview map which can be observed alongside a detailed view on a smartphone.

Embedded Sensors

While the ideal method for controlling content on a HWD remains an open problem, many interesting possibilities are presented through the availability of embedded sensors. One such possibility is speech recognition, used by Google Glass [73] to present and respond to a menu of available voice commands. In cases where interactivity is minimal, speech or context-based interaction can allow hands-free operation. For instance, water usage can be displayed beside a sink when it is used, or a jogger's heart rate can be continuously displayed while she is running. One potentially useful interaction mode that remains to be thoroughly explored is the use of head-tracking. By combining gyroscopic readings with the forward camera view, the device can sense where a person is directing their attention, be it toward virtual content or toward people and objects in the real world. This can be used to facilitate context-oriented interactions, such as presenting a virtual business card alongside a colleague’s face or activating visualizations related to particular objects. It is also possible to embed devices with eye-tracking sensors to enable more precise gaze-based interactions. Sensors that track hands can enable ordinary surfaces to become interactive touch surfaces, allowing the use of standard gestures such as tapping for selection, flicking for scrolling or pinch-to-zoom. In-air gestures are also possible with floating displays while on the go or when touch interaction is impractical, for instance while following a messy baking recipe. Proxy objects can potentially be tracked using computer vision or network-connected inertial sensors to allow other forms of input such as raycasting with a stylus or virtual cursor manipulation using a mouse.

Requirement	Description	HWD Opportunities
Mobility	Supports analysis in the environment or situation where the data are collected or applied	HWDs and wearable input devices and can be used while at home or work, or while on the go. HWDs can support hands-free use
Integration	Information should be integrated into the user's environment through ambient displays or overlaid onto objects in use	Spatial interfaces place content in surrounding space and embedded sensors allow precise alignment for augmenting real-world objects with information displays
Interpretation	Information should be easy to interpret for non-experts and presented in an engaging fashion	Interpretation can be simplified by augmenting objects with information in the correct context. Flexibility of HWD interfaces allows 2D or 3D objects to be placed anywhere to provide imaginative and fun experiences
Multiple Views	Multiple views allow introduction of additional information for overview or comparison. Interpretation can be simplified by distributing multiple simple views instead of a single complex representation	Augmentation allows an unlimited number of displays to be placed anywhere without extra cost. Spatial interfaces spread multiple views in space for fast, intuitive switching
Interactivity	Gaining insights requires exploration of the data through interactive visualizations. Selection and navigation operations should be coordinated across views	Embedded sensors can track gaze, hands and other objects to provide many possible interaction methods. HWDs can work in conjunction with other devices to enable interaction both for manipulating display views and interacting with their contents. Augmentation allows views to be coordinated with interspatial links, while a spatial interface allows users to find the best physical viewpoint

Table 5. Summary of HWD Opportunities
This table summarizes our proposed design requirements for in-situ, everyday analytics, and how the opportunities afforded by upcoming HWDs support each of these requirements.

3.3 Ethereal Planes Metaphor

In our current work we root our interface designs in a metaphor we call Ethereal Planes [58], in which content is placed within a set of 2D virtual windows situated in 3D physical space. In this metaphor, windows act as 'containers' in a similar vein to traditional desktop interfaces; however, the windows in Ethereal Planes are not constrained to the boundaries of a physical display. SAIs leverage several benefits from the situation of these 2D windows in 3D space. For example, spatial memory and proprioception can be utilized to store and retrieve information components. The virtual windows can be manipulated and organized in such a fashion to benefit interpretation, for instance by placing related information sources side-by-side for cross-referencing. Physical space can also be leveraged by placing windows in the vicinity of appropriate objects or by drawing meaningful visual links across intervening space between windows or to connect data points to physical locations.

We differentiate Ethereal Planes from the concept introduced by ElSayed et al. [54] of *situated analytics*, where information is rendered directly onto related objects in the environment. Both SAIs and situated analytics are similar in their use of AR display technology to support in-situ, analytic tasks. However, situated analytics assumes an explicit spatial relationship between the data and the outside world, making it particularly appropriate for particular datasets, for example geographical data. With SAIs, in-situ opportunities may be found without such an explicit spatial relationship, for instance determined by the temporal or opportunistic nature of a given task. The SAI concept also places a greater emphasis on spatial interaction, which leverages body motion, whereas a situated analytic AR interface might be viewed and controlled through the screen of a smart phone or tablet. Furthermore, simple versions of SAIs (e.g. a body-centric array as in **Figure 3**a) do not require the degree of sensing and tracking precision to overlay content directly on real-world locations as is required for situated analytics, and can be implemented using today's technology.

Despite some apparent limitations of a window-based interface, there are several practical reasons why we choose the Ethereal Planes metaphor. First, even in a spatial visualization of geographic or other spatially-related data, it is easy to imagine cases where additional window interfaces would be useful. For example, if one is viewing the paths of people's movements projected onto the floor of an environment, the analytic task may benefit from a map showing the same paths in a top-down overview of the entire space. Many applications could further benefit from flat panels containing abstract controls or text. Also, windows act as containers for organizing and compartmentalizing information. This organization helps to prevent virtual information from cluttering or obscuring important objects in the real world that may lead to unwanted or dangerous distraction. Furthermore, there is evidence that 2D visualizations are more clearly interpreted and can be more easily manipulated than 3D visualizations. Finally, 2D interfaces are familiar to users and can, in some cases, incorporate existing applications or familiar elements.

3.4 Conclusion

This chapter discusses a variety of technologies that can be used to as a platform for SAIs. We single out HWDs as an ideal technology for this type of interface and outline many of the specific benefits afforded by this technology. Although the five design requirements for SAIs are drawn from research based on more traditional interfaces, we can adapt these to the unique properties of HWDs. These requirements are drawn from work aimed at supporting analytic tasks to support sensemaking, and are not specifically related to any particular technology. For instance, when an analyst pieces together information from multiple views, these views may be spread across multiple display monitors, or distributed across a single large display. While each platform may have particular strengths and weaknesses in supporting each requirement, the use of any particular technology is not a requirement in itself. Furthermore, our sources for compiling these requirements include works aimed at adapting visualization and visual analytic techniques to personal technologies [53,100,168]. In this chapter, we discussed why HWDs have a strong potential for providing such a personal platform. Nonetheless, care must be taken to adapt these requirements for the benefits and limitations of new technologies such as commercial-grade HWDs, which have yet to be tested with analytic interfaces such as SAIs.

4 Mobile Multi-Window Layouts

In this chapter, we explore two of the primary SAI requirements, **mobility** and **multiple views.** For an interface to be used in-situ, in a wide variety of contexts, mobility is an essential feature. One way to implement a mobile interface on a HWD is to situate 2D visualization windows in space. Such a design also depends on a second important requirement, multiple views. To ground our design, we draw from our Ethereal Planes framework the concept of egocentric (body-fixed) window layouts. With an egocentric layout, a set of user-defined application windows will maintain their relative positions to the user as the they move, making this a practical approach for mobile context. The goal of this chapter is to explore the ideal design of such a multi-window layout for mobile users, which we call the Personal Cockpit.

We explore egocentric window layouts in a cascading series of user studies, with each building on the results of the previous studies. Through these studies, we explore a number of design parameters, such as window distance from the user, angular width, (i.e. apparent size) of windows and angular separation between windows.

We also take a preliminary look at **interactivity** with window content. In particular, we explore direct input, which we use to mimic touch, as is used in modern touchscreen interfaces. Interaction techniques based on 'natural' human interactions with real world objects have advantages, such as learnability and discoverability for novice users, as well as allowing a transfer of knowledge from currently popular devices. However, computer interfaces, and virtual environments in particular, also allow designers to provide new abstract interaction techniques that in some cases may be more productive than natural ones, as well as 'hypernatural' interactions that supersede 'natural' ones. We explore a greater range of interaction methods, including a combination of natural and abstract interaction techniques, in Chapter 6.

We include an exploration of spatial reference frames as a design factor in our studies, to determine whether a body-fixed window arrangement is the most practical for direct input. Due to the human-factors focus of these studies, we conduct our studies in a CAVE environment, which emulates the FoV limitations of a HWD. This low latency, high-fidelity environment provides us with data of high quality from which to draw conclusions, without confounding influences of current HWD technology.

A final user study in this chapter provides a preliminary validation of our spatial analytic concept for HWDs. In this study, participants conduct an analytic task that draws from information spread across multiple windows. The results confirm those of similar studies on other spatial interface platforms which inspired our concept for SAIs; participants complete the task more efficiently by window-switching via head motion than with two baseline view-fixed interfaces we implemented. Also, the majority of participants indicated a preference for the Personal Cockpit interface, finding it to be 'more natural' than the view-fixed techniques. Our primary finding is that navigating our spatial interface layout using head motion is by far the fastest, despite the imposed FoV limitations as would be experience with a see-through HWD.

4.1 Related Work and Design factors

We explore the design space for an interface ideally suited for multi-tasking on HWDs, the Personal Cockpit. Our work is inspired by a number of interfaces that leverage spatial memory to bridge the gap between real and digital worlds. Much of this work can be traced back to Fitzmaurice's information spaces [65], which map information to its associated physical locations in the real world. Feiner [63] later implemented a HWD interface with virtual windows mapped to world- and body-based reference frames. In Billinghurst's following work [21,22], we see the potential of head-tracking [200] for improving interaction with multiple displays. Many similar world- and body-centric concepts followed on other platforms such as spatially aware mobiles [176,230] and projectors [36]. We build on these prior works by pinpointing relevant design issues that we use to guide our design process.

Field of View (FoV)

We are interested in how FoV limitations impact the Personal Cockpit. The human visual field spans about 200° horizontally and 130° vertically, however the detail-oriented foveal region of the eye spans only about 3° [161]. Information from different regions is split between two physiological pathways, called the ventral and dorsal cortical streams [161]. The ventral pathway, related to pattern recognition and fine detail, processes information primarily from the fovea. The dorsal region processes information from both the central and outer regions and is related to motion sensing and navigation. Tasks that require such information from the peripheral view are known to be particularly susceptible to effects caused by FoV limitations [161], for instance finding a walking path through a room [3] or avoiding obstacles [206].

In addition to direct effects on physical tasks , a wide FoV contributes to a user's sense of 'presence' in a virtual environment [123,127]. See-through HWDs, however, allow wearers to retain their peripheral view of the real world; a limited FoV affects only the 'augmented' content provided by the device. Also, reliance of the dorsal cortex is lessened in certain tasks, such as reaching, when the target is constrained to the central foveal view region. Since precision of the reaching movement [169] primarily depends on foveal capture of the intended target, a wide angle display may not be necessary for object manipulation tasks. Other research indicates that, for information processing tasks relying mainly on the ventral cortical stream, a FoV of only 40° may suffice [161].

The available FoV on current low-cost HWDs varies between about 23° (e.g. Moverio [62]) to 40° (e.g. Laster [116], Lumus [129]). The impact of FoV on performance is also gender dependent [49].

Context Switching

Multiple displays have benefits over single displays for multitasking, particularly when display switching costs are minimized. Dual monitors can reduce workload and task time for frequent switching [106,190] and much information can be safely relegated to a secondary display [190,224]. Cauchard et al. [38] studied display separation in a mobile environment and found that context switching does not drastically impair performance of a visual search task, provided that head movement is minimized. Rashid et al. [171], however, found visual search to be slower when split between a mobile and large display than on either display alone. Spatial constancy in multi-window layouts can improve memorability and reduce switching time [178,201].

Angular Separation

Given that the Personal Cockpit requires head movement, we consider the effects of angular separation between the multiple displays. The range of human neck motion for a normal adult is relatively large: about 85° for rotation to either side, 50° for vertical flexion (looking down) and 60-70° for vertical extension (looking up) [135]. However, the effective range for task switching is smaller. For example, Su and Baily [195] found that two displays on the same vertical plane can be displaced by up to 45° before incurring negative effects on a docking task.

Display Size or Display Angular Width

Display size can influence task performance, although the effects are dependent on viewing distance. When viewing distance is held constant, we refer to display size as angular width. Ni et al. [154] found that large, high resolution displays improve performance on navigation, search and comparison tasks. Shupp et al. [186] found that large displays benefit certain tasks, such as search, but not others, such as route tracing. Ball and North [16] argue that the affordance of physical navigation has a greater effect on task performance than display size. Similarly, physical motion could prove advantageous for multitasking.

Window Distance

As our design of the Personal Cockpit includes direct user input, window distance is a primary design factor. For virtual displays, the impacts of depth are numerous. The minimum comfortable distance of binocular convergence is about 0.25m [213], although ergonomics research recommends placing desktop monitors at a distance of at least one metre [7]. Tan and Czerwinski [202] found that performance is negatively impacted by mixed display distances. Thus our Personal Cockpit design should keep the working set of windows at a single depth. Estimation of depth is known to be impaired in virtual environments [38,161,206], due in part to FoV restrictions [229].

A well-understood phenomenon and cause of simulator sickness is vergence-accommodation mismatch. This effect occurs when the proprioceptive cues of focus and vergence become decoupled in stereoscopic environments [161,184]. Until this issue is circumvented by technological advancements (e.g. [97]), HWD designers can reduce unpleasant effects by keeping the depth of virtual objects close to the surface of the virtual image plane [184,213].

One further design consideration on HWDs with limited FoV is binocular overlap. As illustrated in Figure 12, the viewing frusta of both eyes typically overlap exactly at the distance of the display's virtual image plane. A device can be designed to allow a wider FoV by only partially overlapping the frusta. This choice comes at a trade-off in performance [161] due to monocular regions on the sides of the viewing region. Binocular overlap is also reduced when a large virtual object appears wider than the available viewing region. For example, the lower window in **Figure 12** is cropped to a different region for each

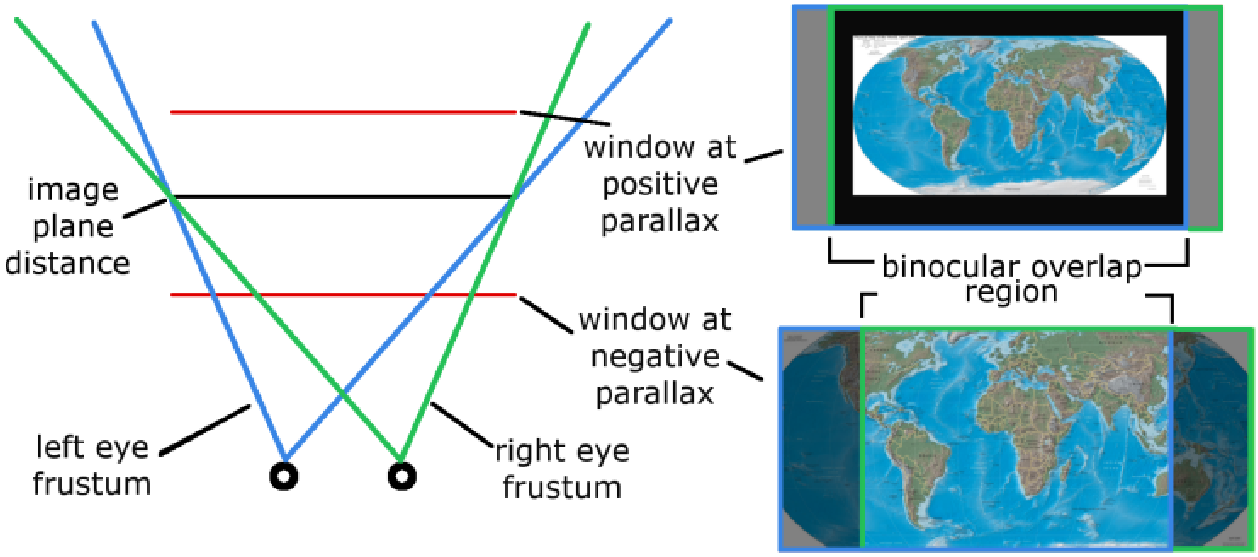

Figure 12. Effects of Binocular Parallax on HWD View
Binocular parallax creates an illusion of depth in stereo images (left). If objects appear in front of or behind the head-worn display's (HWD's) virtual image plane, the binocular overlap region is reduced (right). Content appearing wider than the available overlap region results in regions to either side that can be viewed by only one eye (bottom right).

eye. One particular item of interest we explore is how the distance of a virtual display affects the interpretation of its contents.

Direct Input

Whereas the direct manipulation metaphor allows intuitive interaction with virtual objects [185], our Personal Cockpit design must take into account several issues inherent to 'touching the void': Depth perception of virtual objects is difficult and the depth estimation of a virtual surface is made more problematic by the lack of a tangible surface [40]. Furthermore, when distance is overestimated, the user's penetration of the surface can cause double vision, or diplopia [40,210]. Also, interactive objects must remain within average maximum reach, about 50-60 cm to the front and 70-80 cm to the dominant side [152].

HWDs present additional challenges for direct input. In a wearable system, head-tracking and registration relies on body-fixed sensors. Thus, robust tracking and motion stabilization are required to create a convincing illusion of spatially situated objects. Also, since the display is located physically between the viewer and the locations of situated objects, a virtual display will occlude the user's hand as it reaches the surface. To make direct input feel natural, the system should detect the reaching hand and make it appear to occlude the virtual display. We circumvent these issues in our studies by emulating a HWD in a CAVE setting, in which case the display is behind the user's hand.

Spatial Reference Frames

A layout of multiple displays can be classified according to the spatial reference frame to which the displays are fixed (e.g. [22,63]). For example, user elicitation study on organization of multi-display layouts [81] resulted in both environment-centric and user-centric layouts. We can similarly affix virtual displays to objects or location in the physical world (*world-fixed*) or to some part of the observer's body, such as the head (*view-fixed*), torso (*body-fixed*) or hand (*hand-fixed*). Because HWDs are easily portable, we explore the impact of different reference frames on direct input with the Personal Cockpit.

Display Layout Curvature

Shupp et al. [186] explored the differences between curved and flat layouts of multiple monitors when aligned to form a large display. Their study shows that performance with search and route tracing tasks are 30% faster on the curved layout than the flat layout. One advantage of the curved display revealed in this study is that the region of focus can be changed easily by moving one's head, whereas the flat layout requires users to shift their body. A curved layout with oblique views at a uniform distance from the user may likewise support efficient task switching on a HWD; a closer viewpoint will provide a greater pixel resolution of the virtual displays, and the reduced need for body motion will benefit users in a virtual environment as well as a physical one. Furthermore, a curved layout is well suited for reaching with an extended arm. Accordingly, we use a curved layout for the Personal Cockpit.

4.2 User Studies

We refine our design of the Personal Cockpit as an advanced interface for multi-tasking on HWDs through four user studies. In the first three studies we fine-tune the design parameters (Figure 13) of display size (angular width), distance, reference frame and angular separation. These studies explore the human factors that underlie the design of a multi-window layout on a HWD, so we evaluate performance metrics such as task time, pointing error, perceived effort and perceived fatigue. In the last study we compare the Personal Cockpit against navigation techniques commonly used on smart phones, adapted for task switching on view-fixed HWD interfaces. The goal of this study is to determine whether natural, spatial interaction using head motion supports more efficient task switching than 'standard' abstract navigation methods, based on metrics of time and user preference. As a basis for this study, we use an analytic task consisting of multiple, separate information displays, however, at this stage we focus on the efficiency of task switching, rather than evaluating the quality of analytic taskwork.

Emulation Environment

As we focus on human-factors limitations in our design, we run our studies in a projection-based CAVE environment. The low display latency and high-precision optical tracking enable us to explore designs not practical on current hardware, to control for confounding background clutter and to examine previously untested design factors.

We emulate the restricted FoV of a HWD by clipping the viewing frusta of users to 40° × 30° (all windows in our studies have a 4:3 aspect ratio). We chose a 40° width because this angle is thought to be sufficient for detail-oriented tasks [161] and is within the range provided by currently available stereoscopic headsets (e.g. Lumus, Vuzix). As with actual see-through HWDs, the FoV restriction only affects virtual content; the real world view remains unobstructed.

To facilitate direct input, we explore visual output within reach of the user. However, many devices have a virtual image plane distance of 3m or more (e.g. Moverio, Vuzix), which is impractical for use within reaching distance. Some devices have an adjustable image plane distance (e.g. Laster SmartVision), supporting objects in the near field. We emulate an image plane distance (**Figure 12**) of 1 m, about the expected limit for use with direct input [184,213]. As with FoV, this choice serves as a worst-case setting in which we evaluate the human-factors aspects of our design.

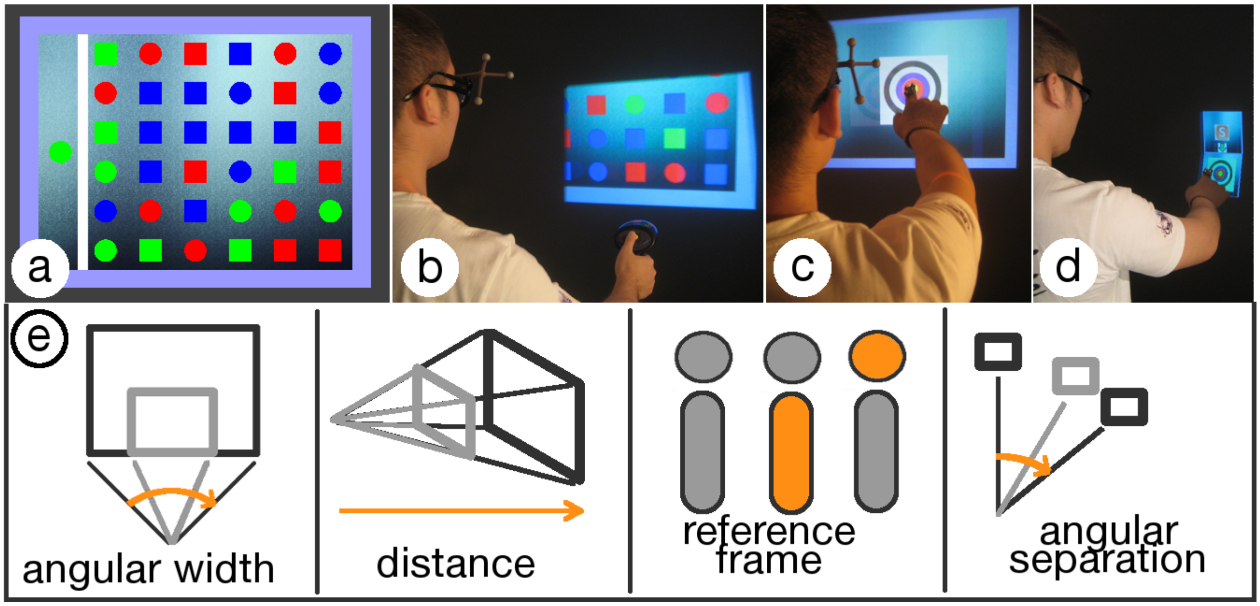

Figure 13. Personal Cockpit Study Images and Design Parameters
a) The stimulus used in the visual search task in Study 1. b) In some conditions, the emulated FoV restriction prevented the full task window from being viewed at one time. Participants entered their response using a wand device. c) Study 2 required users to select targets placed at different depths using direct input. d) Study 3 explored target selection between multiple displays in horizontal and vertical directions. We used the results of our first three user studies (a-d) to tune the design parameters (e) of the personal cockpit, including display angular width, display distance (depth), spatial reference frame, and angular separation between displays.

Our environment does not take all possible issues into account, for example vergence-accommodation mismatch (see Window Distance, above) or the problem of unwanted hand occlusion (see User Input, above and Transferring the Personal Cockpit to a HWD, below). However, it allows us to examine issues related to FoV restriction such as the effects of binocular overlap and the efficiency of navigating to displays that are hidden out of view.

Study 1: Single-Display Viewing Parameters

Our first study explores size and distance placement for a virtual display. These values depend on the FoV and distance limitations for direct input. Displays that appear wider than the FoV width are not fully visible from a single head position and may be difficult to interpret due to a reduction of the binocular overlap region (see Display Depth, above). We expect participants will be more efficient when the virtual display's angular width is equal to or less than the FoV. Participants may also prefer virtual displays that appear further away (i.e. with a lesser offset from the virtual image distance plane).

Participants, Task and Procedure

We recruited 10 university students (two female, 21 ≤ age ≤ 40 years) from our local campus. We screened participants using a binocular depth test, which required them to differentiate between virtual displays placed at close (60 cm), intermediate (100 cm) and far (140 cm) distances. As a result of this test, we had to turn away two participants.

We implemented a visual search task to examine the effects of display width and distance. We use a conjunction search [208], in which the target and distracter objects share multiple properties. In our case, objects can share the same shape (square or circle) or colour (red, green or blue). The display is partitioned by a vertical line, with a target object appearing on left (**Figure 13**a-b). The participant must search an array of randomly generated objects on the right side of the line and count the number with the same shape and colour as the target object. Participants report their count by pressing one of four buttons on a handheld wand device. Virtual displays appear directly in front of the participant, centred at eye-height. Participants are asked to complete the task as quickly and as accurately as possible.

Design

We use a 5 × 4 within-participants design. The factors are *angular width* of the virtual display, relative to FoV (*50*, *75*, *100*, *125* or *150%)* and apparent *distance* of the window (*40*, *60*, *80* or *100* cm). Conditions are presented to participants in random order to reduce learning effects. Within each condition, participants complete ten trials consecutively. To measure performance, we record trial time and the number of incorrect selections. We collected 5 angular widths × 4 distances × 10 trials × 10 participants = 2000 data points. After each set of ten trials, participants provided perceived *effort* (on a 7-point scale) by answering the question "How hard did you have to work (mentally and physically) to accomplish your level of performance?"

Results

We analysed data of recorded trial times and subjective scores of overall effort. In this study and those that follow, we remove outliers greater than 2SD from the mean.

Trial Time: We removed the first trial from each set (200 trials) to eliminate learning effects. We removed further 50 trials (2.78%) as outliers. The mean time of the remaining trials is 3.065 s (SD 1.157 s). We ran the univariate ANOVA for our analyses. Mean times for *angular width* and *distance* are shown in Figure 14. Results show a main effect of *angular width* ($F_{4,36.03}$=58.863, p<.001), but not distance ($F_{3,27.04}$=.106, p=.956). Post-hoc comparisons with Bonferroni corrections show significant differences between all *angular width* conditions (p<.001) except for 50 vs. 75 % (p=1.0).

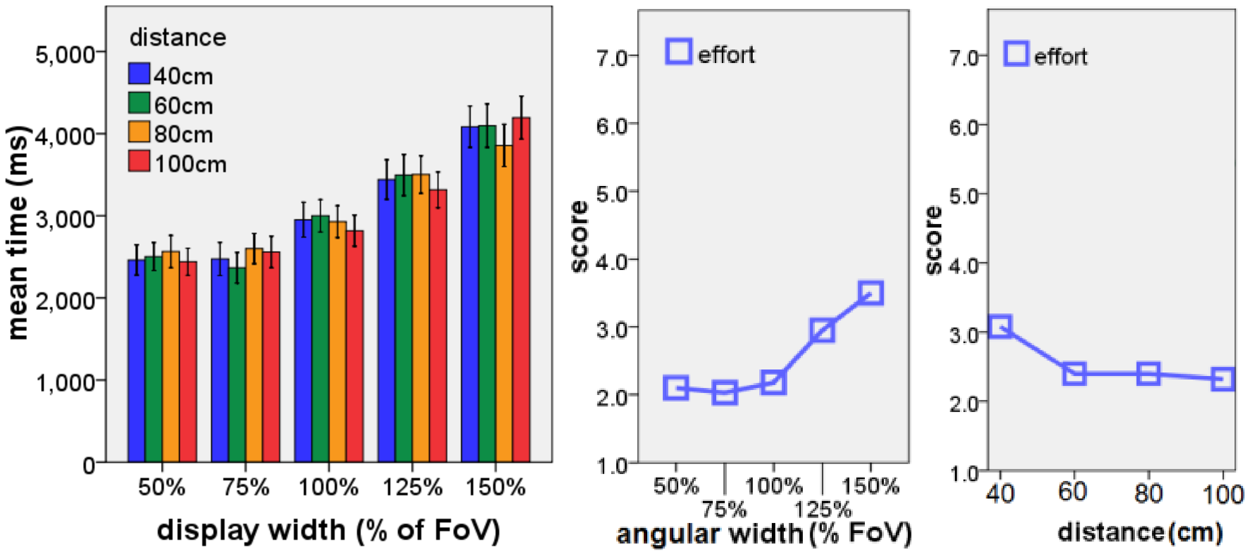

Figure 14. Study 1 Results: Trial Time and Effort
Mean trial times by *angular width* and *distance* (left). Mean *effort* for *width* (centre) and *distance* (right) conditions. Bars show ±2 SE.

Effort: Participants provided scores after each condition for their perceived level of overall effort. We ran Friedman's ANOVA tests for each factor followed by post-hoc Wilcoxon tests. We found an effect of effort on both *angular width* ($\chi^2(4) = 63.44$, $p < .001$) and *distance* ($\chi^2(3) = 22.15$, $p < .001$). Mean scores are shown in **Figure 14**.

Discussion

We find that task time is directly influenced by the ratio of the display width to FoV. Task time is optimal when the virtual display is roughly ¾ the size of the FoV, likely due to reduced head motion. We see a small change from 100 to 75%, but no improvement with the smaller 50% ratio. Interestingly, perceived effort scores, in response to display width, follow an identical pattern to task time. We find that participants perceive increased discomfort at the nearest display distance (40 cm), but task performance is unaffected by distance. This result leaves open the possibility for direct input, as this latter factor is limited by the user's reach.

In the following study, we explore direct input for the average reach of 40-60 cm. As per our findings, in the remaining studies we restrict the window to be approximately ¾ of FoV width to fit completely within the FoV.

Study 2: Single-Display Input Parameters

Whereas Study 1 focused on visual output, Study 2 explores direct input. Our first goal is to determine which display distances best facilitate target selection. Our second goal is to see how the choice of spatial reference frame affects input that relies on reaching. In combination with Study 1, we can determine the ideal balance of design parameters to support both output and input. We expect that participants will benefit from proprioception with body-fixed or view-fixed windows, leading to lower targeting error.

Participants, Task and Procedure

We recruited 12 university students (two female, $21 \leq age \leq 35$ years). From a resting position, participants were asked to quickly and accurately 'touch' the centre of a 10cm diameter bullseye target with their right hand (**Figure 13**c). The target is placed at one of five locations on a virtual window. Based on the outcome of the previous study, we chose a window width smaller than the FoV (70%). The target provided colour feedback to indicate correct or incorrect (the display is penetrated outside of the target region) selections. Participants began the next trial by returning their hand to the resting position. Input detection is provided by a Vicon tracking system.

Design

We used a 3 × 3 within-participants design. The factors are: spatial *reference frame* (*world-fixed*, *body-fixed* or *view-fixed*); *distance* of the display (*40*, *50* or *60* cm) and target *location* (*centre*, *top*, *bottom*, *left* or *right*). Body-fixed and view-fixed displays appeared at a set distance from the participant's body, as determined by the *distance* condition. World-fixed displays are initially set at the same distance, but are fixed to world coordinates and do not move with the user. *Distance* and *reference frame* are presented in a random order to reduce learning effects. Within each condition, participants complete five blocks of trials. Within each block there is one trial at each *location*, presented in random order. To measure performance, we record trial time and target selection error. Participants provide ratings of perceived *fatigue* for each combination *distance* and *reference frame* by answering the question "What was the level of fatigue from this task?" We collected 3 distances × 3 reference frames × 5 target locations × 5 trials × 12 participants = 2700 data points.

Results

We analysed task completion *time*, pointing *error* and subjective ratings of *fatigue*. We found no effects of time, so focus our reporting on the remaining metrics.

Pointing Error: We define error as the distance between the detected selection and the target centre. For error analysis, we included all correctly completed trials. We compared error distances using a 3 × 3 × 5 univariate ANOVA. We found main effects of *distance* ($F_{2,22}$=4.443, p<.05), *reference frame* ($F_{2,22}$=13.759, p<.001) and *location* ($F_{4,44}$=4.780, p<.005) on pointing error. Post-hoc comparisons with Bonferroni corrections show significant differences between all pairs of *distance* ($p \leq .017$) and *reference frame* ($p \leq .003$). Mean pointing error distances are shown in **Figure 15**. There was also a significant interaction effect between *distance* and *location* ($F_{8,88}$=3.762, p=.001).

Fatigue: Participants rated *fatigue* on a 12-point Borg scale. As the Borg CR10 [155] scale was designed to be a linear mapping between perceived and actual intensity, we treat the resulting scores as scalar, using a univariate ANOVA. Results, as shown in **Figure 15**, reveal a significant effect of display *distance* ($F_{2,22}$ =13.162, p<.001). However, we did not find an effect of *reference frame* ($F_{2,22}$ =1.152, p=.334).

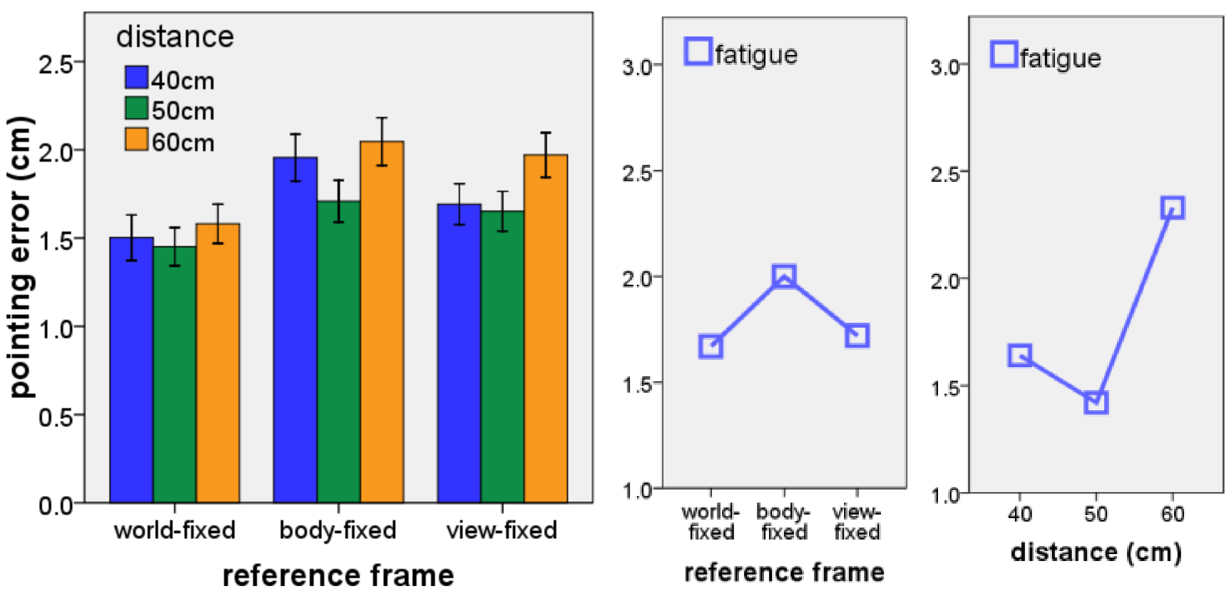

Figure 15. Study 2 Results: Pointing Error and Perceived Fatigue
Mean pointing error by *reference frame* and *distance*. (left). Mean perceived fatigue (rated using a linear Borg scale, from 0-10) levels by *reference frame* (centre) and target *distance* (right). Bars show ±2 SE

Discussion

We were surprised to find that target selection is clearly more precise in the world-fixed reference frame. Any benefits of proprioception in the other two reference frames were overshadowed by unintentional motion of the target window caused by the pointing motion. Although distance did not influence pointing speed, there was an unexpected effect of distance on pointing error. This effect was strongest in the body-fixed frame, i.e. when the window moves with the body, likely due to the unintentional window motion. Error was greatest at 60 cm, where participants' arms were near full extension. Precision was particularly bad in the top and left target locations, which required a slightly greater (right-handed) reach.

Study 3: Multi Window Layouts

Having refined the distance parameter for direct input and visual output, we now investigate layouts of multiple windows, with target selection between two windows. Study 1 showed the best task performance when the window fits fully within view. Multiple tasks, however, are likely to occupy separate windows that span beyond the user's FoV. The ideal placement range is limited by human factors including the range of neck motion for a typical user and performance of direct input. As study 2 showed negative effects on pointing error from even subtle body motions, we use a world-fixed frame for optimal input. We choose a curved window layout for this study to keep targets within reach. However, a curved layout has a natural focal point. To determine if windows are best centred directly around the user, or offset to the side of the dominant pointing arm, we include *focal point* as a study factor. The *centre* focal point is symmetrical to the participant whereas the *right* focal point coincides with the right shoulder. All windows are placed at an equal distance (50cm) from the current point of focus. Multiple windows are offset radially by a given *separation angle* (**Figure 13**e).

Participants, Task and Procedure

We recruited eight university students (two female, one left-handed, 21 ≤ age ≤ 35 years) from our local campus. Participants are presented with a two small windows (**Figure 13**d). One window contains a start button and is placed at shoulder height directly in front of the focal point (centre or right-offset). The second window contains a bullseye target, and is displaced either horizontally or vertically from the start window. The participant begins by 'touching' the start button, then moves quickly and accurately to the target.

Design

We use a 4 × 5 × 2 within-participants design. The factors are: *direction* of display displacement (*up*, *down*, *left* or *right*); displacement *angle* (*15°, 25°, 35°, 45°, 55°*) and point of *focus* (*centre* of body or *right* shoulder). For each *focus*, participants complete ten consecutive blocks of trials, where one block contains all combinations of *direction* and *angle*. Trials in a block are presented in random order to prevent learning effects. The order of *focus* presentation is balanced between participants. We collected 4 *directions* × 5 displacement *angles* × 2 points of *focus* × 10 trials × 8 participants = 3200 total trials.

Results

Trial Time: Time is measured from the moment the start button is tapped until a selection is detected. For analysis of trial completion time, we included only correctly completed trials (i.e. the target selection falls within 5 cm radius of the target centre.) We removed the first trial from each condition (320 trials) to reduce learning effects. We removed a further 88 trials (3.15%) as outliers. The mean time of the remaining trials is 0.70 s (SD 0.27 s). Mean trial times are shown in **Figure 16**. A univariate ANOVA reveals main effects of *direction* ($F_{3,21}$ =7.252, p<.005) and *angle* ($F_{4,28}$ =86.107, p<.001), but not for *focus*. Post-hoc tests with Bonferroni corrections showed significant differences between all pairs of *angles* and *directions* (all p<.001) except for *up* vs. *left* (p=1.0). There was also an interaction effect

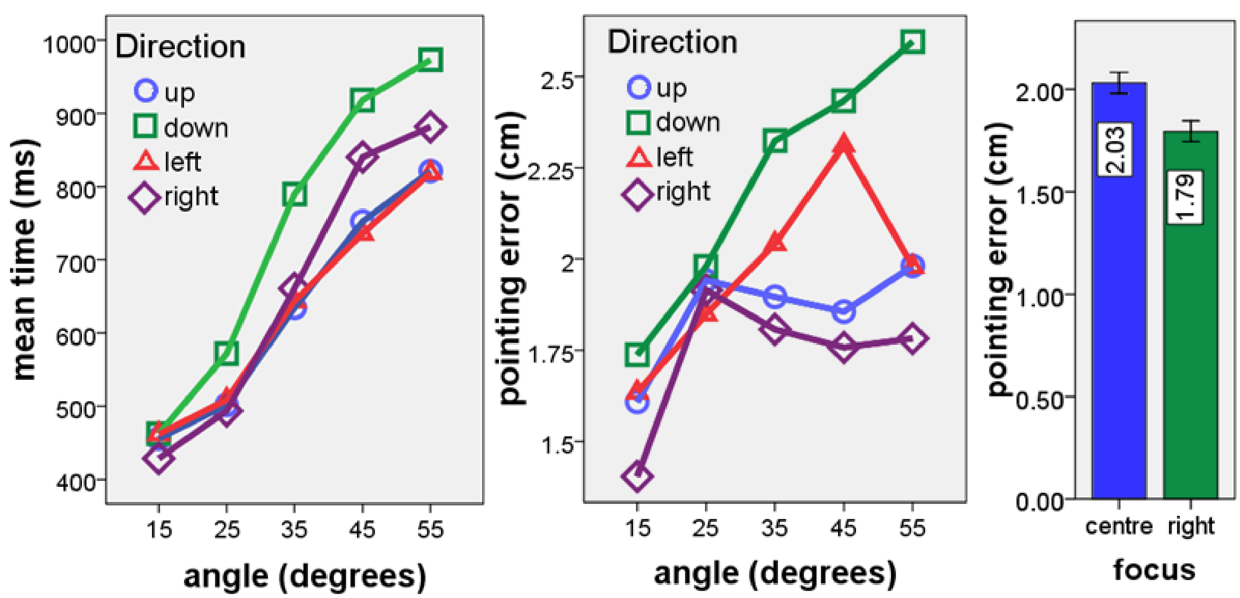

Figure 16. Study 3 Results: Trial Time and Pointing Error
Mean trial times by *direction* and separation *angle* (left). Mean pointing error by *direction* and separation *angle* (middle). Mean pointing error by point of *focus* (right). Bars show ±2 SE.

between *direction* × *angle* ($F_{12,84}$ =3.579, p<.001) as well as a 3-way interaction between *direction* × *angle* × *focus* ($F_{12,84}$ =2.678, p<.005).

Pointing Error: As in study 2, error is the distance between the detected selection and the target centre. For error analysis, we removed 138 outliers (4.31%). Mean values are shown in **Figure 16**. A univariate ANOVA revealed main effects of *direction* ($F_{3,21.003}$=4.115, *p*<.05), *angle* ($F_{4,28.010}$=6.290, *p*<.001) and *focus* ($F_{1,7.002}$=21.204, *p*<.005). We also found a significant 3-way interaction for *direction* × *angle* × *focus* ($F_{12,84.156}$=2.816, p<.005).

Fatigue: Since this study requires both arm and head motion, we collected Borg ratings for both *arm* and *neck fatigue*. Due to the high number of conditions, we grouped separation angles into two *groups*, *low* (15°-35°) and *high* (45°-55°). We collected ratings for all combinations of *direction*, *group* and *focus*. We ran a 4 × 2 × 2 univariate ANOVA for each set of ratings. For *arm fatigue*, the test revealed significant effects of *direction* ($F_{3,21}$ =4.734, *p*<.05), *group* ($F_{1,7}$ =15.465, *p*<.01) and *focus* ($F_{1,7}$ =5.984, *p*<.05). Neck fatigue showed only main effects of *direction* ($F_{3,21}$ =5.500, *p*<.01) and *group* ($F_{1,7}$ =13.213, *p*<.01). Results are shown in **Figure 17**.

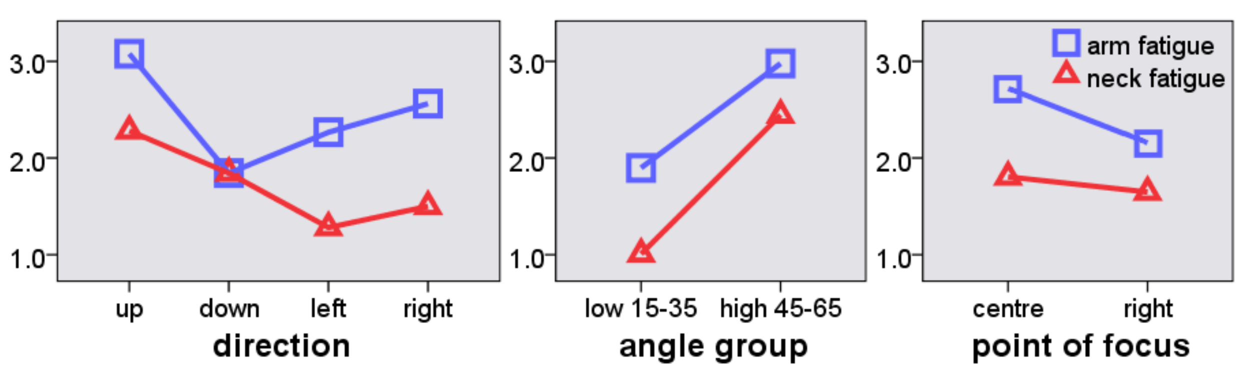

Figure 17. Study 3 Results: Perceived Fatigue
Mean perceived *arm fatigue* and *neck fatigue* by *direction* (left), *group* (centre), and *focus* (right).

Discussion

Time and error are both higher for targets in the *down* direction than for *up* (**Figure 16**). Despite this finding, several participants preferred the *down* direction to *up*, as it reduced arm fatigue (**Figure 17**). Pointing time generally increases with angle, as expected, due to increased travel distance. However, there is a steep increase in around the 35° mark, when the start button and target both fit barely within view. Although *focus* doesn't affect pointing time, there is a significant reduction in error when the centre of curvature is shifted to align with the right shoulder. As a result of this finding, we explored various options for right-offset layouts before implementing the final study.

Study 4: Display switching

The goal of this final study is to demonstrate that the Personal Cockpit, tailored based on the above set of results, facilitates effective task switching over current methods of application switching on HWDs. Whereas our first three studies explored subsets of the overall design space through abstract studies, we designed a more ecologically valid task for this fourth study.

Personal Cockpit Layout

We envision the Personal Cockpit as a versatile, configurable window manager that will be useful for many scenarios, including on-the-go multitasking. However, since study 2 showed that body-fixed windows are prone to target error, we use a world-fixed reference frame for our study. To keep windows within easy reach of the user, we chose a curved layout for the Personal Cockpit (**Figure 18**). Using the best input/output distance from studies 1 and 2, and the right-offset from study 3, we place each windows 50cm from the user's right shoulder. To keep a 4 × 4 array within head range [135], we use a separation angle of 27.5°. To prevent window overlap, we reduce their width to 22cm (60% of FoV at 50 cm distance). Once the window position is determined, we set each window's orientation to face the user's point of view. Finally, based on results from study 1, we correct the window viewing distances. Since the right-shoulder focus causes some of the windows on the user's left to be placed uncomfortably close, we displace windows along the line of sight so each is a minimum of 50 cm viewing distance (**Figure 18**).

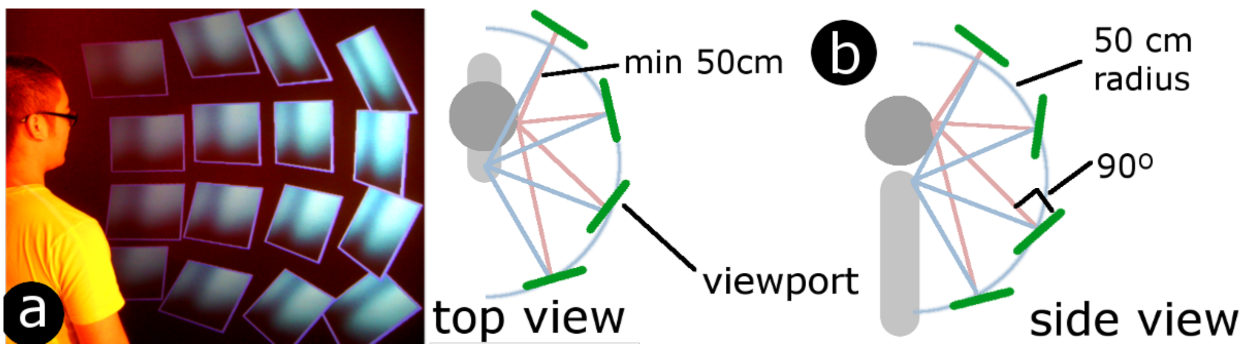

Figure 18. Personal Cockpit Final Design
Our final design of the Personal Cockpit (a, b) was based on findings from studies 1-3, and was used in our final study. Displays are reduced in size to fit fully within the given FoV, and set at a depth of 50 cm for best use with direct input. The arc of display positions is focused on the user's dominant arm to maintain a common reaching distance, with each display oriented toward the user's face for orthogonal viewing.

Participants, Task and Procedure

We recruited 12 university students (three female, 21 ≤ age ≤ 40) from a local campus. Participants are presented with a set of windows showing everyday applications, representing ones that might be used on a real HWD. The goal is to scan the windows for information needed to answer a question (**Figure 19**). The windows present all of the information required to select the correct answer, thus the participant must navigate *between* windows, but need not pan or scroll within the applications themselves.

An example task goes as follows: the participant begins a trial by pressing the button on the Start window, triggering the appearance of icons on the Question and Map windows. The participant navigates to the Question window to find out who he is meeting. Next, he finds the message next to that person in the Messages window. It looks like he is meeting for pizza, so he navigates to the Map window to locate the pizza icon marked with the letter ‘a’. Finally, he returns to the Question screen to select the correct answer, ‘a’, which ends the trial.

There are two question types, one with four applications (Start, Question, Messages, Map), as in the example, and a second type that requires the participant to navigate five

Figure 19. Study 4 Analytic Task
Example of the application windows presented to participants in study 4. This analytic task requires participants to piece together information from multiple views to answer a question.

applications (Start, Question, My Contacts, Calendar and Map). The applications are randomly placed among empty desktop windows within an array of either 9 or 16 windows. The windows are laid out in space according to our Personal Cockpit design and the user switches applications by moving his head (**Figure 20**a).

In addition to our Personal Cockpit design, participants must navigate using two baseline techniques with view-fixed displays: one with direct input and the other with indirect input (**Figure 20**b-c). In these techniques, the same application windows are arranged in a flat array, but the participant can only see those that fit within the 40° FoV. With the direct input technique, the user switches applications by panning the intangible, view-fixed surface (**Figure 20**b). This technique is analogous to panning on a typical smartphone. To assist direct input, we provide visual feedback to indicate whether the reaching finger is above, on, or behind the window surface. Based on previous work showing difficulties with depth judgement [38] and pilot testing, we provide a substantial 'surface' depth of 8 cm.

The indirect technique uses a wireless trackpad, with which participants control a cursor on the view-fixed display (Figure 20c). To switch applications, the participant must select a home icon at the bottom of the display, which leads to an overview of the entire array (c, inset). From the overview, he can select any window in the array, which brings the corresponding window into full view. This technique is similar to the menu interface on some existing HWDs (i.e. Moverio). For consistency, all application windows are sized to 22cm width and placed at 50 cm viewing distance for both view-fixed techniques.

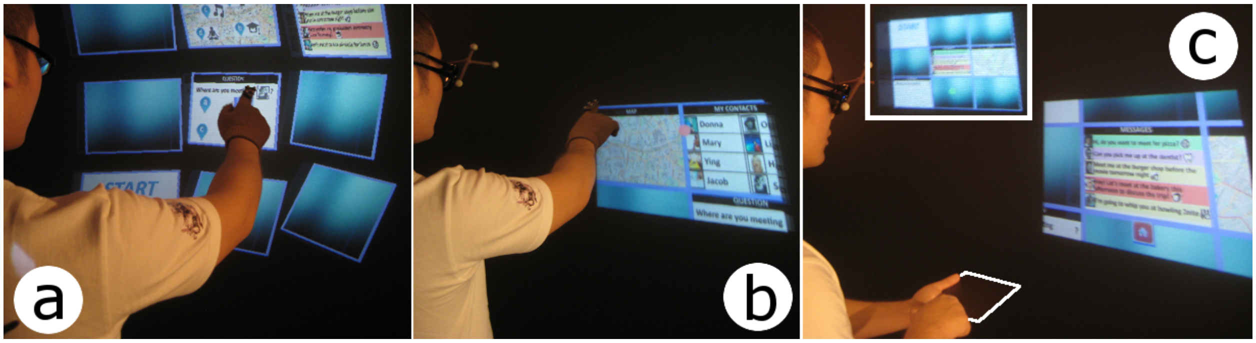

Figure 20. Study 4 Techniques
Study 4 tested our design using head motion against two baseline techniques using abstract navigation techniques with view-fixed displays. a) our final Personal Cockpit window layout design, without FoV constraint in for demonstration. One baseline technique used direct input for panning (b), and the second technique used indirect input on a trackpad (c) to navigate to and from a home screen.

Design

We use a 3 × 2 × 2 within-participants design: *technique* (*PC*: Personal Cockpit with direct input*; VD*: view-fixed with direct input; or *VI*: view-fixed with indirect input); *complexity* (*3 × 3* or *4 × 4* array of virtual windows) *and question type* (type *I* or *II*). Within each technique, participants completed four sets of questions, one for each combination of *complexity* and *question type*. For each new set, applications were moved to new random window locations, but with a minimum of one application for each row and column in the layout array. Each set of four questions was completed using the same window layout. *Techniques* and *complexities* were fully balanced between participants. Type I questions always preceded type II.

Results

Trial time was measured as the duration between the task start and the selection of the correct answer. We collected 3 techniques × 2 complexities × 2 question types × 4 questions × 12 participants = 576 data points. Of these we removed 24 outlier trials (4.17%). The mean time was 19.91 s. Conditional means are shown in **Figure 21**.

Participants completed the trials significantly faster ($F_{2,22}$=94.845, *p*<.001) using *PC* (mean 13.57 s) than either of the view-fixed techniques (23.73 s for *VD* and 23.45 s for *VI*, **Figure 21**). Post-hoc tests with Bonferroni corrections showed significant differences between techniques *PC* vs. *VD* and *PC* vs. *VI* (both pairs p<.001), but not between *VD* vs. *VI* (p=.547).

The univariate ANOVA revealed significant effects of *complexity* ($F_{1,11.187}$ =39.937, *p*<.001) and *question type* ($F_{1,11.051}$ =11.143, *p*<.01). The simpler *3 × 3* complexity had a mean

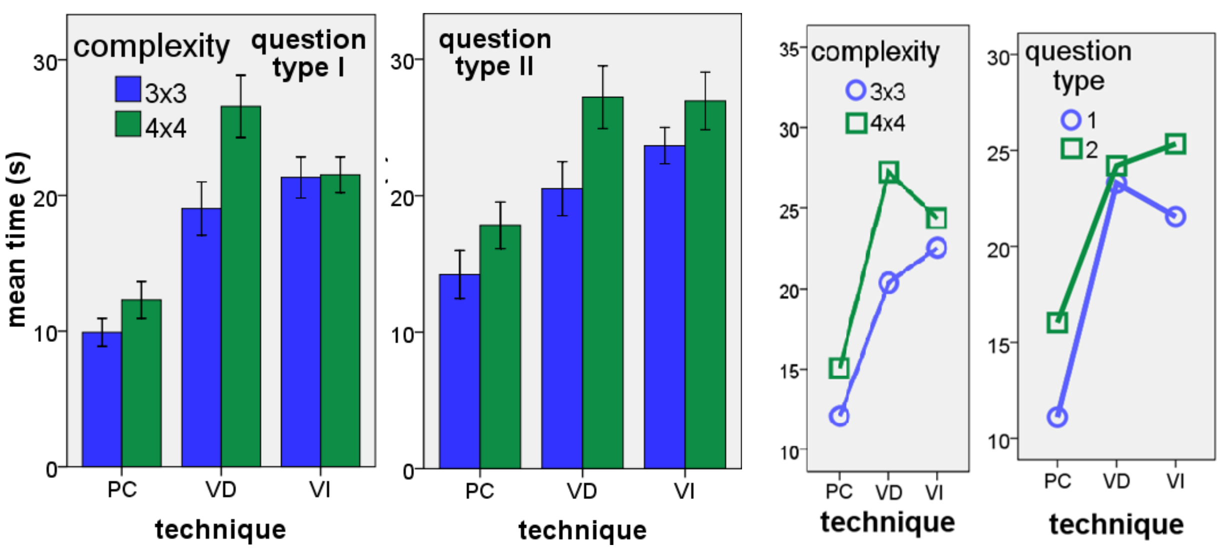

Figure 21. Study 4 Results: Trial Time
Mean time for study 4 by *technique*, *complexity* and *question type* (left). Interaction effects by *technique* × *complexity* and *technique* × *question type* (right). Bars show ±2 SE.

time of 18.33 s while the *4 × 4* trials averaged 22.15 s. Question type *I* was also faster than type *II* (18.65 vs. 21.82 s). We also found interaction effects (**Figure 21**) between *technique* × *complexity* ($F_{2,22}$ =5.976, p<.01) and *technique* × *question type* ($F_{2,22}$ =3.747, p<.05).

To measure fatigue, we collected subjective ratings of arm fatigue and neck fatigue for each combination of *technique* and *complexity*. Means are shown in **Figure 22**. We ran 3 × 2 univariate ANOVAs for both arm and neck fatigue. For arm fatigue, the test revealed significant effects of technique ($F_{2,22}$ =22.045, p<.001) and *complexity* ($F_{1,11.090}$ =7.510, p<.05). Post-hoc tests with Bonferroni corrections show differences between *PC* vs. *VD* and *VD* vs. *VI* (p<.05 for both pairs). There was also a significant interaction between *technique* and *complexity* ($F_{2,20}$ =2.761, p<.05). For neck fatigue, there was a main effect of *complexity* ($F_{1,11.168}$ =8.822, p<.05) but not technique ($F_{2,22.405}$ =7.334, p=.055).

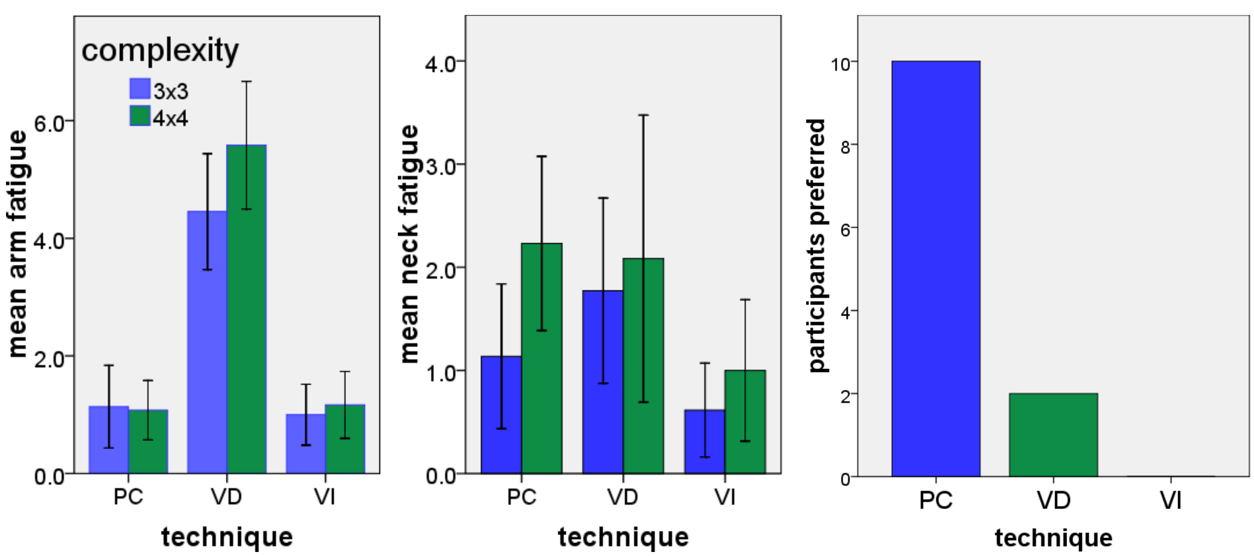

Figure 22. Study 4 Results: Perceived Fatigue and Technique Preference
Mean Borg scale ratings for perceived *arm fatigue* (left) and *neck fatigue* (middle). Ten of twelve participants (83.3%) preferred the Personal Cockpit technique, using head motion, over two baseline techniques using abstract navigation with view-fixed displays (right). Bars show ±2 SE.

Discussion

Our final study shows potential for the Personal Cockpit as an alternative to interaction with view-fixed displays. Our technique is more efficient than both tested view-fixed techniques and less tiresome than direct input on a view-fixed display. Of the 12 participants in our study, 10 chose the Personal Cockpit as their preferred technique in a post-study questionnaire (**Figure 22**). Several participants commented that the Personal Cockpit was "*easy to navigate*". One participant said, "*I liked the speed of navigation - I was able to move around quickly and in such way it reduced the amount of work.*" Others mentioned that it was "*productive*" and "*the most natural*".

The Personal Cockpit is also scalable. Whereas the panning technique (*VD*) shows a large increase in time with a greater number of application windows (**Figure 21**), the Personal Cockpit shows only a small increase, as with the indirect method (VI). Despite the use of direct input and necessity of head motion, participants rated the Personal Cockpit on par with the indirect interaction technique (*VI*).

Our results are positive but come with some limitations. We tested only 2 baseline techniques. Although faster untested navigation techniques may exist (e.g. joystick or trackball), these may not have all of the advantages of the Personal Cockpit (i.e. unoccluded forward view; facilitates both navigation between windows *and* interaction with window contents). Further study with additional tasks is required for generalization, however our results are in line with those of prior research [21,38,121]. Further studies with actual HWD hardware are required for ecological validity.

Personal Cockpit Interactions

To further illustrate the utility of the Personal Cockpit, we created several interactive demonstration concepts. We implemented these in our CAVE emulation to show how the Personal Cockpit might be useful in real-life scenarios.

Window Overview: Although the Personal Cockpit user can access many applications quickly and easily, there may be times when an overview (also known as a 'World in Miniature' in VR [193]) of all open windows is useful. With a command gesture, the user can shrink the Cockpit layout into a palm-sized sphere (**Figure 23**a), which fits easily into view. Attached to the non-dominant hand, the user can manipulate the sphere for convenient viewing.

Changing Frames of Reference: The Personal Cockpit is as mobile as the HWD device and can be designed to follow the user on the go with a body-fixed reference frame. When at work or at home, the Cockpit windows can be fixed to a wall or other available space [130,164]. In this demo, a user can switch between a flat, world-fixed layout and a curved, body-fixed layout with a tap on the HWD (**Figure 23**b).

Manual Arrangement: Our Cockpit design in Study 4 demonstrates a customized automatic layout. Depending on the situation, the user may want to rearrange the windows manually. In this demonstration, the user can grab, move and resize windows at his leisure using in-air pinching gestures. To open a new application window, the user grabs an icon from a task-launcher window and places it in an empty location, where a new window springs into view (**Figure 23**c). A hand-fixed reference frame is convenient for bimanual interaction techniques.

Window Intercommunication: In multitasking situations, two or more windows may be tied to the same task. For instance, many tasks can have peripheral information or tool palettes in a secondary display. We demonstrate a colour-picker tool, in which the user can select a colour to tint a photo in a separate application window (**Figure 23**d).

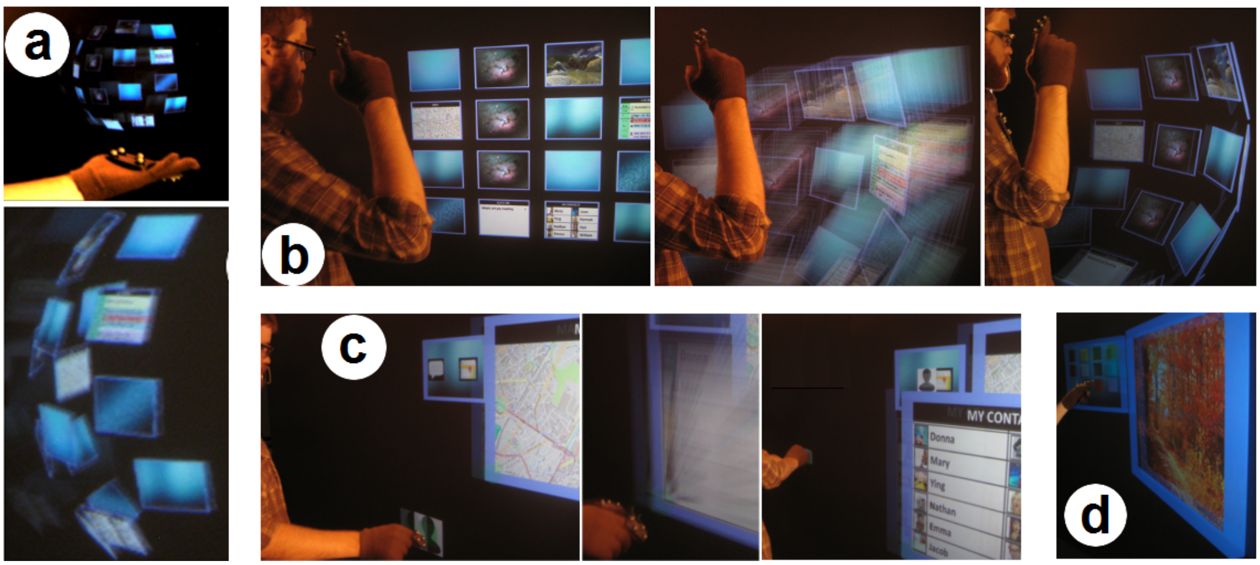

Figure 23. Personal Cockpit Interaction Scenarios
We implemented several examples of potential interactions using a Personal Cockpit interface to manage layouts and applications: a) changing from world-fixed to body-fixed layout; b) opening a new application window; c) window intercommunication; and d) shrinking the Cockpit to a palm-sized overview.

Translating the Personal Cockpit to a HWD

Our next step is to demonstrate that Personal Cockpit's design advantages transfer to a real-world HWD. As display and tracking technologies advance, systems will be able to support fully-mobile implementations. We outline some important challenges for this realization.

Body-Fixed Stabilization: In Study 2, we found that naïvely fixing windows to body-fixed coordinates leads to selection inaccuracies with slight body motions. Based on this finding, we envision a hybrid between world- and body-fixed reference frames for mobile use. When the user is standing still, the layout becomes purely world-fixed. When he begins moving again, the system detects this and brings the Cockpit along. Other approaches include using a low-pass filter to smooth and stabilize motion.

Pseudo-Occlusion: An important problem we discussed earlier (Direct Input, in Design Factors section) is that a HWD lies between the viewer and the input space. This causes the display image to occlude any outside objects, including the user's hands. We propose the concept of *pseudo-occlusion* to solve this. The system would accurately track the hands' position in nearby space. When the hand is placed between the HWD and a virtual window, the system subtracts the interfering region from the rendered window, making it appear occluded by the hand.

Transfer to Wearable Technology: Our emulation of the FoV limitation in a CAVE environment provided us with several advantages in implementation and tracking. Further research is required to discover the limitations of applying a functional Personal Cockpit interface on current and near-future hardware with variations such as different image plane distances. We also must answer questions about the effectiveness of transparent displays in real situations, such as with objects moving in the background or while walking.

4.3 Conclusion

We take away the following lessons from our investigation:

1) The spatial multi-window layout of our design allows fast task switching, requiring only 60% of the time of the 2 tested view-fixed interaction techniques.

2) Virtual windows are compatible with direct input, even with a limited FoV. Windows can be placed as close as 50 cm, even with a 1 m distant virtual image plane.
3) Body-fixed reference frames are subject to higher targeting error than world-fixed windows, due to unintentional perturbations caused by reaching motion.
4) A curved layout is subject to lower error and arm fatigue when offset to align with the dominant limb.
5) The Personal Cockpit is scalable within reasonable limits. Greater window offset angles, and thus greater window numbers, lead to increased head motion. This can lead to longer task switching times.

5 World-Integrated Window Layouts

In this chapter, we primarily explore the **integration** of displays into the built environment. Whereas the previous chapter explored egocentric (body-fixed) window layouts to maximize mobility, this chapter focuses on exocentric (world-fixed) window layouts. In particular, we explore how to help integrate an SAI into the user's surroundings by aligning windows with existing surfaces. Our goal is to minimize the intrusiveness of the interface, making it suitable for ambient applications, or to coexist alongside existing objects (e.g. kitchen appliances) or interface components (e.g. desktop computer), and make information fit seamlessly into the user's current activities. We call this interface design the Personal Façade [60,61].

The Personal Façade attempts to preserve a user's familiarity with the spatial layout of a set of application windows by preserving its relative configuration. To help the windows integrate seamlessly into the environment, it also avoids occluding important objects in the scene. The Personal Façade's layout manager balances multiple weighted constraints to find a candidate layout from the given search space. While a wide range of constraints are possible, the primary constraints we explore are spatial constancy and visual saliency. The spatial constancy constraint maintains the relative spatial relationships between windows as they might appear in a Personal Cockpit layout, including their given order and positions relative to the user. This keeps layouts consistent between different rooms (**Figure 24**), to support **mobility** between various environments. The visual saliency constraint prevents windows from occluding important objects detected in the user's view to minimize intrusion of the interface and maximize the potential for smooth integration into the surroundings.

To understand the importance of the Personal Façade's constraints, consider the following scenario: A busy executive sitting in his office (**Figure 24**a) quickly glances at an agenda he knows is on the desktop to his lower-right; before initiating a call, he rapidly locates a meeting place on a map found on a wall above, to his upper-right. Later, while sitting in his living room (**Figure 24**b) he hears a notification. He instinctively looks to his lower-right, where he again finds his agenda; however, this time it appears on a nearby seating surface. To refresh his memory of his upcoming meeting location, he looks up to the map application, once again positioned on an adjacent wall to his right. These actions are done using natural motions and spatial memory, and do not require holding a device or navigating through multiple application layers.

5.1 Related Work

The concept of window managers for assisting task organization can be traced back to early developments of personal computers, and their incorporation into spatial user interfaces (SUIs) occurred early in the history of virtual reality [64]. With the introduction of see-through head-mounted displays, researchers such as Feiner et al. [63] and Billinghurst et al. [22] imagined multiple windows being anchored to different objects in the environment or arranged in body-centric configurations. Later work recognized the potential of a 3D spatial environment for leveraging spatial memory to assist the recall of items [1,177]. Computer users have been shown to be extremely adept at using spatial memory to find previously seen items, however this ability requires that items remain spatially constant [130,164,182,201]. The application of spatial constancy to location recall has received little attention in the context of SUIs, despite foundational developments making it possible in AR applications (i.e. registration) [115].

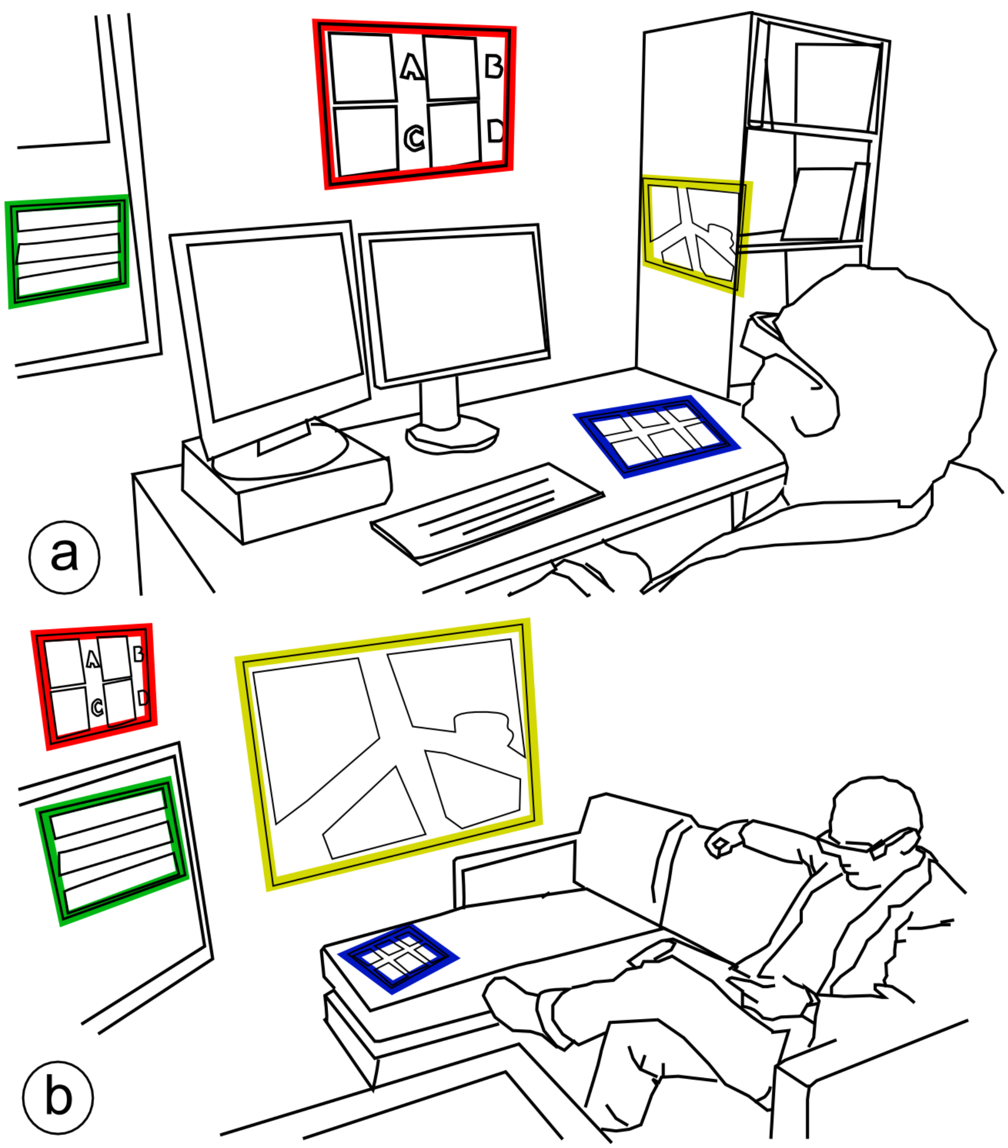

Figure 24. Personal Façade Spatial Layouts
The Personal Façade integrates multiple applications into surfaces in the user's surroundings. Application window layouts remain spatially consistent in different environments (e.g. a) an office and b) living room) while avoiding occlusion of important scene objects.

In addition to optimizing spatial constancy, the Personal Façade layout manager takes into account surface geometry and background visual appearance for determining the placement of application windows in the surrounding environment. The exploitation of surface geometry is of potential benefit to HWD interfaces, for instance to improve content legibility by mitigating dual disparity [114] or to provide a tangible input surface. However surface structure has been explored primarily in the context of projection-based interfaces, which explicitly require a projection surface such as the workplace walls in the visionary Office of the Future [173]. The introduction of portable handheld projectors led to systems that dynamically adapt to the environment's surface geometry [36,172]. An early goal of these systems was to correct distortion for legibility. Such perspective correction can also be used when the observer is mobile, for example in multi-display environments [150].

Surface detection has also inspired the development sophisticated 'immersive room' environments, in which projection surfaces encompass entire walls while maintaining awareness of objects within the room [102,214,215]. Surfaces can be detected dynamically to allow projection onto moving surfaces such as paper or people's hands [176,226]. Advanced prototype systems consisting of sensors and projectors have been developed to simultaneously map the environment and support projection-based interactions [147]. Surface detection has also been incorporated into AR interfaces through the exploration of low-cost techniques such as vanishing line detection [74,119]. In contrast, the Personal Façade assumes the existence of a complete spatial model, which may in future be routinely stored and made available on demand.

In contrast to surface geometry, issues of interference with a display's background have been primarily explored in the realm of augmented and mixed reality. These applications require thoughtful placement of content with respect to the real-world background, particularly on see-through HWD screens, on which foreground content cannot be made fully opaque. For example, to mitigate the negative effects of background texture and luminosity on text legibility [70,120], researchers proposed text colour and contrast adjustments [70] or algorithms to move text to an optimal region of the display for readability [120]. Researchers further elaborated on such techniques by repositioning content dynamically for a moving background [159,160] or by considering components such as background colour [92,203] or visual saliency [74]. Our layout manager is the first to our knowledge to combine both visual saliency and 3D geometric constraints within the same implementation.

In relation to window management, little research has been done to specifically provide efficient access to multiple applications in SUIs. Bell and Feiner [18] introduced an efficient algorithm for dynamically keeping track of available space. One particular work that is closely related to ours, describes the implementation of a window-manager for multi-projector displays, wherein windows are arranged to maximize the available projection space [219]. However, unlike our method, this layout manager does not maintain spatial constancy of application windows as it is not concerned with changing environments. Other work on HWD interfaces however has explored dynamic content placement, for example preventing occlusion of important objects [19]. Some early research explored the concept of attaching application windows to objects using fiducial markers [50,63]. The Personal Façade determines layouts dynamically using only information extracted form camera images and a mesh model.

5.2 The Personal Façade

The Personal Façade is a multi-application management tool for stereoscopic HWDs. Its main component is a window layout generator that embeds virtual 2D application windows in the environment using camera image and depth sensor data. Automatic layouts are created at run time based on the user's current position and orientation, and take into account the geometry and layout of the room. Manual operations are also provided to manually configure layouts for analytic multitasking.

Window Layouts in the Environment

The Personal Façade's layout generator uses a variety of heuristics such as surface geometry and visual saliency to determine where to place application windows in the user's environment. Following is a list of the heuristics that might be considered by a content manager for stereoscopic, see-through HWDs. Below, **Table 6** provides a summary of these heuristics along with a list of prior implementations that have considered each heuristic. This list is not comprehensive but shows how our implementation fits within the current state of the art.

Surface structure

Indoor environments contain an abundance of flat, smooth surfaces, which are ideal for placing 2D content. Additional structural considerations are the size and shape of a given region, its 'orthogonality' (the facing direction of a window relative to the user's view), and its location, including relative direction and visibility.

Background appearance

Many regions in an environment will contain important visual information that should not be occluded. Visual saliency algorithms (e.g. [33]) model regions of a scene that are visually important to human observers. Additional semantic information can be used to identify highly important objects such as faces and text [19,39]. In addition, transparent displays are susceptible to visual effects of texture, colour and luminosity, thus regions that interfere with content legibility should be avoided.

	Heuristic	**Usage**
Surface structure	Surface normal	[102,115,119,214,215], PF
	Size and shape	[219], PF
	Orthogonality	[150]
	View direction	PF
	Visibility	PF
Background appearance	Visual saliency	[74], PF
	Semantic importance	[19]
	Texture	[70,120]
	Color	[92,203]
	Luminosity	[70,120,159,160]
Layout consistency	Spatial constancy	PF
	Window overlap	[219], PF
	Relative order	PF

Table 6. Personal Façade Related Work Overview
Heuristics for placing application content in the environment and prior art that has used each heuristic. Items uses in the Personal Façade are marked with 'PF'.

Layout consistency

A layout manager should make it as easy as possible for users to find information. Spatial constancy, which has been shown to improve task switching time in standard desktop interfaces [201], should be preserved from one environment to another, despite environmental differences in surface structure and background appearance. The arrangement of windows should also minimize overlap and maintain the relative order of windows (i.e. left-to-right and top-to-bottom).

5.3 Implementation

We implemented the Personal Façade using Unity3D on a desktop computer with an NVIDIA Quadro 600 GPU. We created two mock environments for development and testing, made to resemble a typical office and living room (**Figure 25**a,b). Users are able to view the Personal Façade on an Epson BT-100 stereoscopic HWD with 23° diagonal FoV, tethered by a composite video input. We track the HWD using a high precision, low latency Vicon tracking system, thus the virtual content appears through the HWD to be accurately superimposed on the physical environments.

Automatic Layout Generator

Our window layout generator places windows in the mock environments using heuristic-based constraints, such as the layout's spatial configuration and visual salience of the occluded background. A key contribution and component of this layout method is the application of spatial constancy in a real-world spatial layout. However, the concept of spatial constancy in a 3D spatial interface has several possible interpretations. Unlike the fixed space of a display screen, spatial interfaces inhabit various possible coordinate systems, for instance room-fixed [63] or body-centric [22]. Thus, constancy could imply that windows are fixed within the environment, or that they stay fixed relative to the user's body. In the Personal Façade, we take a hybrid approach; we use a body-centric reference frame to keep the layout consistent in different environments. However, we assume the existence of a 'primary' viewing location and direction, which we use to transform the body centric reference frame on to room coordinates. This assumption holds in many real-world environments, for example in an office with a single desk chair. It remains an interesting topic for future work to explore how and when users would opt to update their window layouts as they move about an environment.

Adherence to spatial constancy is further complicated by the visual appearance of the surroundings. The layout generator displaces windows when important background objects are detected, but attempts to minimize such displacement. The primary goal of our layout generator is to perform a balancing act between these opposing constraints to provide satisfactory but efficient layouts.

Our layout generator uses a Monte Carlo approach derived from the Metropolis-Hastings algorithm [88], following from similar implementations that have been shown to be effective for creating constraint-based layouts of objects in space [71,137]. This algorithm evaluates a sequence of proposed solutions, which are incrementally improved on for a fixed number of iterations, with some allowance for random perturbations. Within the evaluation, we define a set of weighted constraints that help us find a suitable layout for the application windows. A constraint is a function that generates a positive score indicating the 'goodness' of a window location (or set of window locations). While a great number of such constraints are imaginable, we used a minimalistic set in our implementation, with the primary constraints defined as follows:

Adherence indicates the location of a window with respect to its location in the default configuration. Windows with a high adherence are obeying the principal of spatial constancy, which makes them easy to find in different environments. The score is calculated as the angle distance of a window's candidate position from its default position, normalized over an arbitrarily chosen maximum angle of 30°.

Non-occlusion measures the degree to which a window is occluding, or overlapping important background information. To quantify this constraint, we measure the background saliency of a region that a window in the candidate position would occupy. High non-occlusion scores are given to windows in regions with low visual saliency.

We also apply constraints taking into account the *View Direction* (to align windows as closely as possible to the user's forward view), the *Surface Fit* (whether a window lies fully in a polygon), users's *Line-of-Sight* (all window corners are in visible locations), *Relative Order* of windows (whether windows maintain their spatial relations e.g. left-of), and *Overlap* (whether windows overlap others). Additional constraints, which we leave for future work could include *Color & Contrast* (choose locations to maximize legibility), *Predictable Locations* (align windows with landmarks such as room corners, wall centers or viewer horizon), *Maximal Size* (choose locations that allow large window sizes) and *Application Context* (by placing windows in locations that best suit the application context [65], for example a clock above a door or a weather report affixed to an outdoor window).

Window Layout Algorithm

The algorithm input consists only of data extracted form a mesh model and a single photo of each environment. The mesh models (**Figure 25**g,h) were made using Kinect Fusion [101] and the photos (**Figure 25**a,b) were taken with a typical SLR camera with a wide-angle lens (110°). In our current implementation we generate a static 3D Model of the scene beforehand, although we envision such a system working in dynamic settings in real time (see Summary and Future Work, below). We begin by searching the vertices of the mesh models for regions of uniform surface normal, from which we extract a set of surface polygons (**Figure 25**c,d) using a greedy search with Hough transforms [187]. Meanwhile, we compute a saliency map of both scenes using the AIM saliency algorithm of Bruce and Tsotsos [33] (**Figure 25**e,f). We chose this saliency method from many available options because of the high contrast and preserved boundaries regions in the saliency map. Finally, we calibrate the 3D model space with the 2D image space of the saliency map using Bouguet's toolbox [28] (**Figure 25**g,h). This provides all of the information needed to enable a rich number of layout options for indoor scenes.

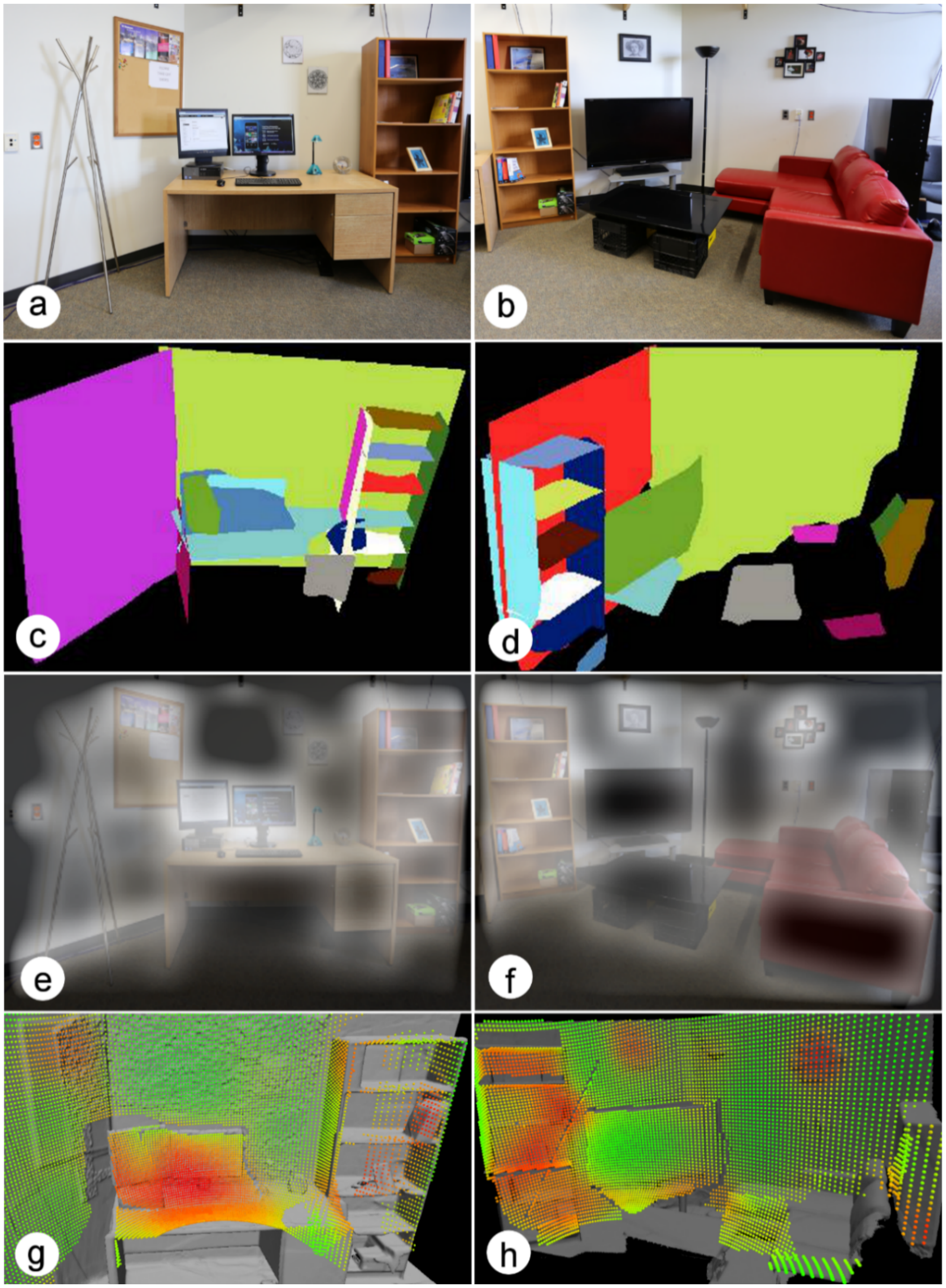

Figure 25. 3D Saliency Map Generation

Our preprocessing pipeline, shown in the office (a) and living room (b) test environments. First, surface polygons are generated from the mesh models (c, d). Next, saliency maps are created using AIM [33]. Light regions are high salience with contrast increased for demonstration). Lastly, saliency maps are projected onto the models using a screen- to model-space transformation (g, h). Red nodes are high salience. A power law transformation is applied to widen the range of scores.

For calculating window layouts, we use a region 90° wide × 45° high, centered on the forward view, discretized into increments of 5°. Since finding the optimal layout L for a set of n windows in a given environment is not currently obtainable at interactive speed, we instead use a Monte Carlo approach derived from the Metropolis-Hastings algorithm [88]. Similar implementations have been shown to be effective for creating constraint-based layouts of objects in space [71,137]. This algorithm evaluates proposed solutions which are incrementally improved over a fixed number of iterations, with some allowance for random perturbations.

First, we define the *layout solution space* as the set of all possible assignments of a set of application windows W to unique points in P_E. We define a 'goodness' function

$$Goodness(L) \coloneqq \sum_i \alpha_i \cdot r_i(L_i)$$

where α_i is an optional weight, $r_i: (O_i \subseteq O) \to \mathbb{R}$ is a constraint operating on a subset of the parameters O, L is a proposed layout solution, and L_i is a subset of the layout containing only the windows with constraints O_i.

The algorithm follows the procedure in **Figure 26**. In each iteration we randomly select a position for one of the windows and re-evaluate the goodness function. We update the solution if improvement was found or with probability p (p = 0.005 in our case). This random factor allows the algorithm to escape local maxima to find better solutions. In our evaluations, we run 2000 iterations of this algorithm to generate an initial solution, then an additional 500 iterations for a 'fine-tuning' phase, in which the pool of possible positions for each window is restricted to within 0.2 m of that in the previous iteration. The primary phase finds a 'good' layout from the whole available space and the fine-tuning phase optimizes that layout within the local maxima. The mean run-time of the procedure in the Unity framework is 3.26 s, however this time can be substantially optimized, for instance by eliminating the mesh model and by cropping to reduce the number of raycasting operations.

Random Walk Algorithm

```
Iterate(∞)
Iterate(0.2 m)
```

Function `Iterate(radius r)`

```
L :=RandomSolution()
bestLayout := L
bestGoodness:= Goodness(L)
for i :=1 to numIterations do
     w := RandomWindow(W)
     position (w):=RandomPosition(P_E,w,r)
     L := L ∩ w
     if Goodness(L) > bestGoodness then
          bestLayout := L
          bestGoodness := Goodness(L)
     else
          if  rand < p then
               bestLayout := L
          end if
     end if
end for
return bestLayout
```

Figure 26. Random Walk Algorithm
The random walk algorithm we use to find window layouts is similar to that of Gal et al. [71]**, which is based on the Metropolis-Hastings algorithm** [88]**.**

Generated Layouts

Some typical outputs produced by the layout generator are shown in **Figure 27**. These outputs are generated using different possible weighting schemas of our constraint functions as shown in **Table 7**. Each promotes a different balance of Adherence and Non-occlusion. The Balanced condition is ideally tuned for the Personal Façade to balance both of these important yet contrasting factors in our test environments (**Figure 27**b). Through trial and error, we found that the Non-occlusion constraint requires a higher weight than Adherence to prevent windows from frequently overlapping high salience regions, such as the area surrounding the desktop monitors in the office setting (**Figure 25**g). Two alternative layout approaches are generated for comparison. The Constancy layout is given a Non-occlusion weight of zero. This theoretically causes each window to be projected onto the nearest surface in line with its default position, however the other constraints and the algorithm's random element cause some deviation (**Figure 27**c). Conversely, the Saliency layout has an Adherence weight of zero. This causes windows to congregate in low salience basins of the environment's saliency map, regardless of their distance from the default location (**Figure 27**d). However, we provide the View-direction function in place of constancy to help prevent windows from moving to extreme distances from the user's forward view.

Layout	**Adherence**	**Non-occlusion**	**View-direction**
Balanced	1	2	0
Constancy	1	0	0
Saliency	0	2	1

Table 7. Constraint Weighting Schemas
This table shows three possible constraint weighting schemas for the Personal Façade's layout generator. Each layout promotes a different balance of spatial constancy and visual saliency. All other weights are set to their default value of 1.

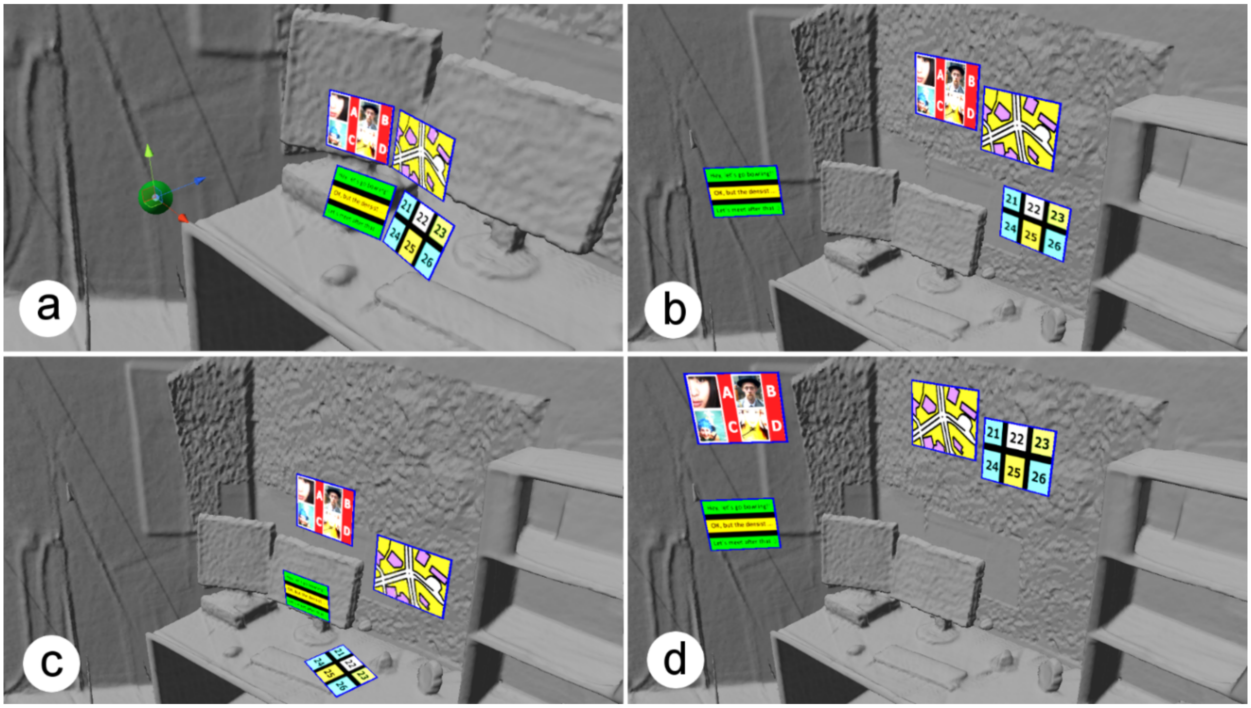

Figure 27. Personal Façade Schema Comparison
Default window locations (a), are set in a 'floating' array, 50 cm from the user's viewing position, indicated by the green sphere. Resulting surface layouts, generated with each of the constraint weights shown in Table 7: Balanced (b), Constancy (c) and Saliency (d). The Balanced layout (b) produces a tighter grouping than the Saliency layout (d), but without occluding the central salient area (i.e. the computer monitors and desk – see Figure 25g), as with the Constancy layout (c).

For comparison, **Figure 28** shows several additional examples of generated layouts. These include four- and six-window layouts in both the office and living room environments. For each combination we show one example of each constraint weighting scheme from **Table 7**. We evaluate these layouts in a user study, described in the following section.

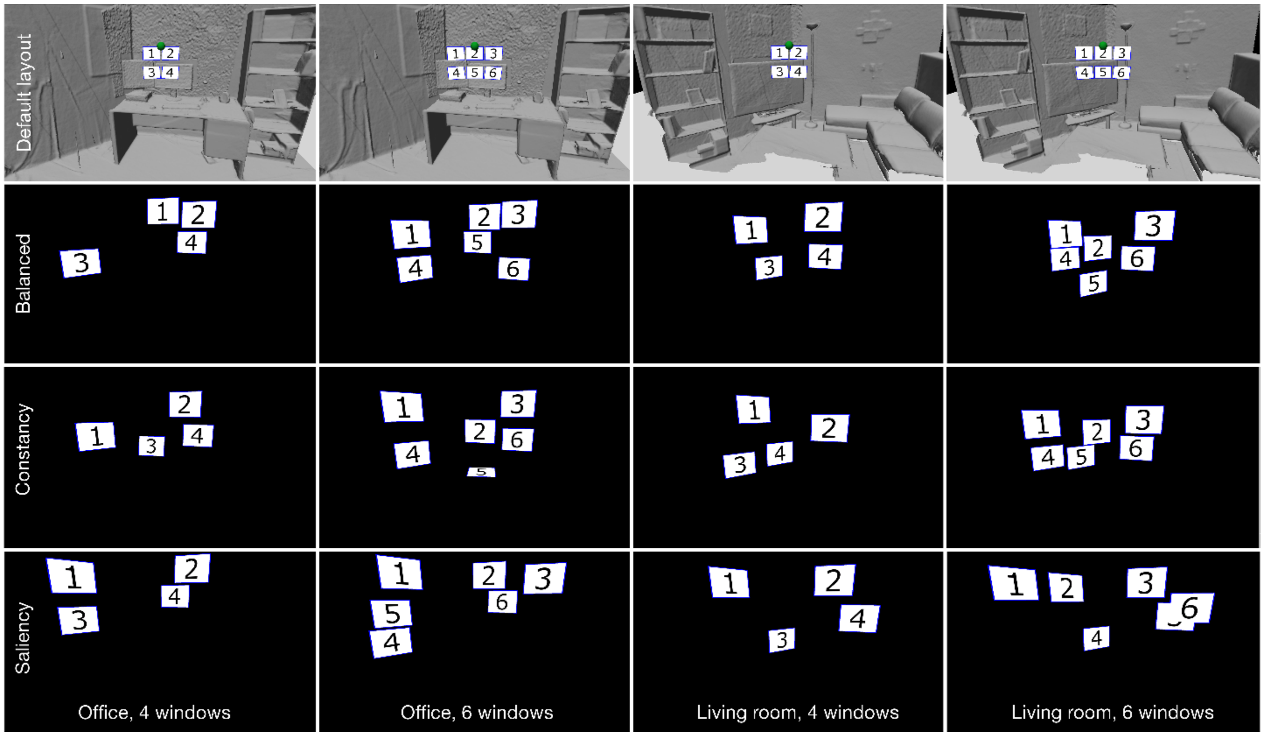

Figure 28. Detailed Schema Comparison
Results of each layout weighting scheme in both environments, with layouts of 4 and 6 windows. The user viewpoint is positioned 1.5 m above the floor, 1.5 m (2.0 m) from the wall in the office (living room) environment.

5.4 Study 5: World-Fixed Layouts

We designed a user study of the Personal Façade layout manager with two objectives: 1) to determine if the layout weighting schemes produced layouts with the intended qualities and 2) to observe real users interacting with the system through a HWD and collect qualitative feedback. To achieve these goals, we timed participants finding windows in the three layout alternatives introduced in the previous section (Balanced, Constancy and Saliency; **Table 7**) and conducted follow-up interviews.

To determine effects of the environment, we conducted the study in both of our test environments described above (**Figure 25**a,b). We arranged these environments to contain different degrees of surface complexity (**Figure 25**c,d) and visual salience (**Figure 25**g,h). The *Office* was denser and more constrained, while the Living Room provided more open space.

Participants

Twelve participants (four female, two left-handed, ages 18-40), volunteered for the study. All participants had normal or corrected vision and were screened for deficiency in perception of color and stereoscopy. All were regular smartphone users and none had previous experience with a HWD nor were familiar with the concept of window layout interfaces with such devices.

Task and Procedure

To probe the effects of spatial constancy and saliency in the Personal Façade's layout approach, we implemented a visual search task, a typical task for querying about the effects of a visual layout on spatial memory [56,182,201]. Our task was composed of two phases. In the first phase, participants scanned a pre-defined layout of six windows, arranged in a body-centric array (**Figure 29**a) similar to that described in the Personal Cockpit interface [56]. Each window showed a randomly chosen three-digit number, with the target window highlighted in light green. This initial phase allowed participants to register the contents of the target window stimulus and the window's relative location in the array. When ready, the participant pressed a button on a handheld wand (**Figure 29**b) to begin the second phase. This triggered the Personal Façade's layout generator, after which

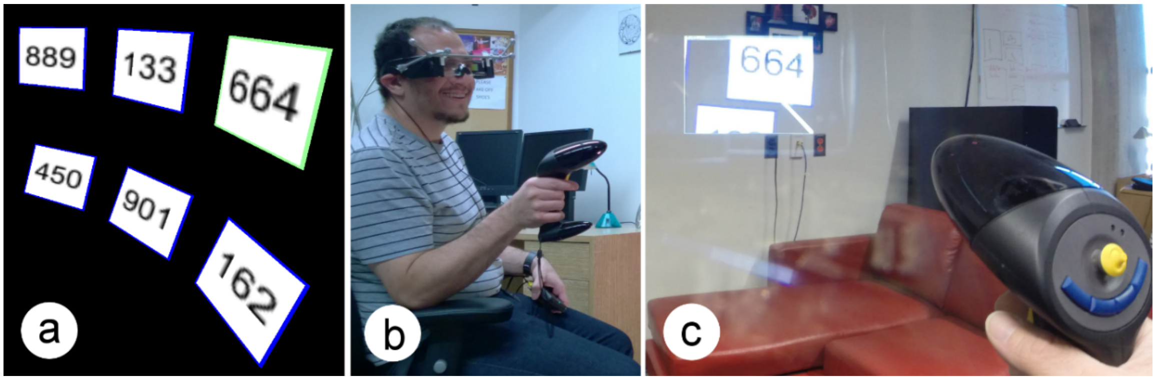

Figure 29. Study 5 Stimulus and Apparatus
The stimulus (a) and apparatus (b) used in the Study 5. c) A window being selected during the search task, as viewed through the HWD.

windows reappeared on surfaces in the environment. After locating the target window, the participant selected it using a virtual ray appearing to extend from the wand (**Figure 29**c).

Participants were allowed to experiment with the interface and get familiar with the apparatus. They were given sufficient training in each environment, and adequate breaks. We asked participants to be as efficient as possible.

The experiment used a 3 × 3 × 2 within-subjects design with the following factors:

- *Layout:* Balanced, Constancy, and Saliency
- *Room:* Office and Living Room
- *Viewing Angle:* Left, Center and Right

We set the viewing position to roughly 1.5 m from the wall in the *Office* setting, facing the desk. In the *Living Room*, the viewing position is farther back, about 2.3 m from the wall, giving a wider, more open view. For experimental validity, we controlled the user viewing position in each environment, however to prevent overly-repetitious layouts, we altered the initial Viewing Angle by increments of 30°. All factors were balanced to mitigate learning effects, with half of the participants starting in either Room. For each Room, participants completed 2 blocks of rotating Viewing Angle within each Layout. Timeouts (30 s) and incorrect selections were requeued at the end of each block, resulting in a total of 432 data points (12 participants × 3 Layouts × 2 Rooms × 3 Viewing Angles × 2 Blocks). We recorded the search time for each trial, measured from the completion of the layout calculation until the target selection.

While the search task allows us to measure layout efficiency, we followed with a second task to gauge the layout quality. In each environment, we showed participants one instance of each Layout in the *Center* Viewing Angle, this time allowing participants to take their time to explore the layout in detail. To quantify the amount of overlap with highly salient objects, we asked participants to count the number of windows covering the objects in cluttered regions of the *Office* (i.e. the desktop monitors, keyboard and mouse). We also collected responses to questions such as which windows were covering objects the participants thought were important and what they liked or disliked about the layout in general.

Apparatus

Content was displayed stereoscopically on a commercially available HWD, an Epson Moverio BT-100. This device has a 23° diagonal FoV, weighs 220 g and has a display resolution of 960 × 540 pixels with a refresh rate of 60Hz. The device measures 17.8 × 20.5 × 4.7 cm. Layouts were computed on the same machine and testing environment described in the implementation section, above.

Results

This section discusses the results of our user study of the Personal Façade. First we discuss the quantitative metrics, of search time and the number of incidences of background occlusion. We then give a summary of qualitative feedback we collected.

Search Time

The mean search time of the 432 successful trials (excluding 4 timeouts and 4 incorrect selections) was 4.11 s (SD 3.31). Mean times for each Layout between Rooms are shown in **Figure 30**a. We applied a log-transform on the non-normal search time data before using a univariate ANOVA. Our analysis showed a main effect of *Layout* ($F_{2,22}$=29.759, p<.001). Post-hoc comparisons with Bonferroni corrections showed significant differences between *Balanced* vs. *Saliency* (p<.001) and *Constancy* vs. *Saliency* (p<.001), but not *Balanced* vs. *Constancy* (p=1.0). *Saliency* was slowest overall (mean 5.07 s), with *Balanced* and *Constancy* taking similar times on average (3.66 s and 3.61 s, respectively).

The mean search time was greater for the *Office* (4.44 s) than the *Living Room* (3.79 s), however the effect was not statistically significant ($F_{1,11}$=4.566, p=.056). There was however a main effect of *Viewing Angle* ($F_{2,22}$=11.930, p<.001) due to differences in the facing surface complexity at different viewing angles. We also found an interaction effect between Layout and Room ($F_{2,22}$=5.693, p<.05).

Overlap with salient objects

We ran Friedman's ANOVA on the reported number of windows overlapping the highly-salient region of the *Office* environment (i.e. the desktop monitors, keyboard and mouse) to look for effects of Layout. We found a significant effect ($\chi^2(2) = 20.591$, $p<.001$) and post-hoc Wilcoxon tests showed differences between all pairs ($p <.05$). Mean counts are in **Figure 30**b.

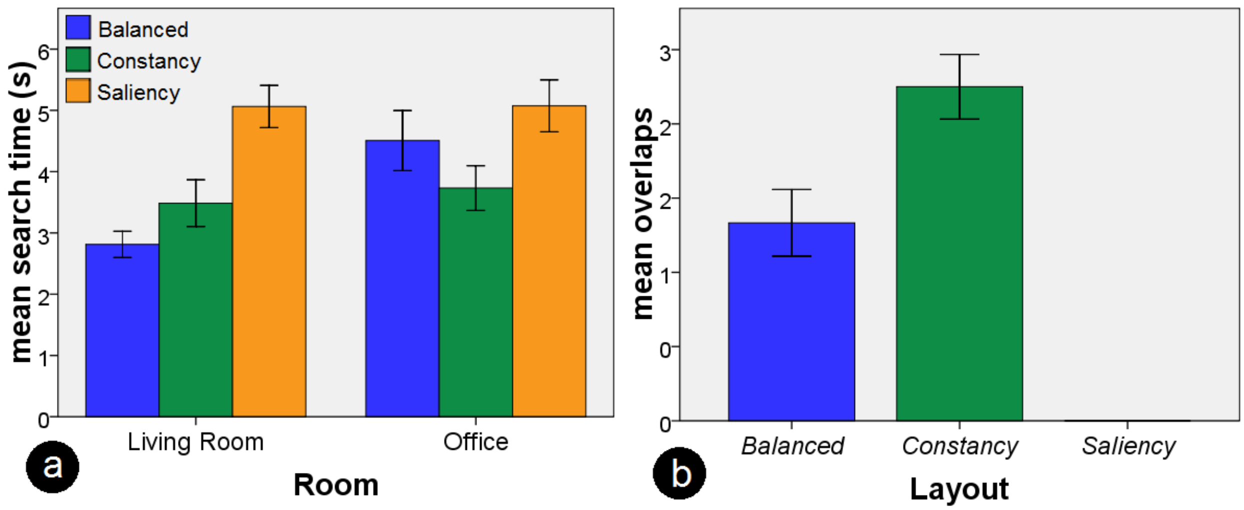

Figure 30. Study 5 Results: Task Time and Scene Occlusions
a) Task time by layout weighting schema for each environment. b) Average number of windows counted that were overlapping salient objects (i.e. the desktop monitors, keyboard and mouse) in the central region of the Office environment. Bars indicate ±1SE.

Qualitative Feedback

Participants provided many insightful comments about the layouts and locations of individual windows. Below we summarize the general trends, which we believe will be useful for informing the design of constraints for future versions of the Personal Façade:

Important scene objects

Participants expressed many individual opinions about what objects should not be overlapped. However, there was a general consensus that the office computer equipment should not be covered, particularly the monitors. Most participants did not like occlusions of the keyboard or mouse, although one participant felt the “keyboard is not important” because he can type by touch, while others were not concerned about the number pad. Conversely, most participants did not mind when windows covered other utilitarian or decorative objects such as the coatrack, books and wall hangings, although some noted it depends on the particular content of the item or situational context (e.g. pictures are acceptable to cover in a work environment). These results highlight the importance of customization in Personal Façade. The interface would benefit from additional knowledge about what objects are important to users.

Context

Window locations can have strong contextual associations. For instance, one participant particularly liked windows on the office desk surface, because it “fits the office paradigm”. One participant considered high windows as “urgent”, while a low window was “ready for the recycle bin” and windows below the desktop monitor were akin to “sticky notes”. Contextual input would help the Personal Façade to determine suitable window placements.

Temporal considerations

Some window locations were not liked by participants because of anticipated future events. For instance, a window covering the living room's TV or wall plugs is not ideal, because those objects might be used. Similarly, a window should not be placed directly above the couch because someone might sit there. Future improvements can enable the Personal Façade to continuously scan environments in real time, allowing such temporal considerations to be incorporated.

Relative Layout

Participants tended to prefer windows in "clusters" as opposed to being "spread out". Similarly, one participant disliked a "big gap in the middle" of the layout. A single window separated from the others was often noted as undesirable because such "outlier" windows could be "hard to find". However, separation between windows was considered acceptable if windows were in groups or even pairs. Several participants said they would prefer if windows were "lined up" with one another or with existing edges in the scene, as opposed to being "staggered". These findings reinforce the importance of aesthetic as well as functional design considerations in Personal Façade's window layouts. We look forward to its future application in different task scenarios to explore how different configurations can best suit user needs.

5.5 Discussion

In general, the different layout weightings produced the results we expected. For instance, the *Constancy* and *Balanced* layouts place windows close to their initial starting position, reducing search time. In contrast, the *Saliency* layout tries to place windows in open flat spaces with low saliency (**Figure 25**g,h), which causes participants to engage in a prolonged search. Although the *Saliency* layout's *View-direction* constraint helps keeps windows near the forward view, however it is seemingly not as effective at reducing search time as the *Constancy* Layout's *Adherence* constraint.

The mean search times for the *Balanced* and *Constancy* Layouts are statistically equivalent, however we observed that the *Balanced* layout is less likely to overlap 'important' objects. Due to the nondeterministic nature of the layout generator's algorithm, windows will occasionally occlude highly salient objects, however less frequently than with the *Constancy* Layout. The results of the Freidman test on Layout support these observations, although we acknowledge the generalizability of this result is limited.

We believe the effect of Viewing Angle was due to a greater complexity in surface structure along the direction of the *Right* Viewing Angle in the test environments. More interesting, however, is the interaction effect between Layout and Room. Although the difference between Room conditions was on the outside margin of significance, we believe differences in these settings played a large role in the observed interaction effect. It seems the conflicting constraints of the *Balanced* Layout sometimes caused windows to be placed in unpredictable locations in the cluttered *Office* environment. Conversely, the avoidance of salient regions in the *Living Room* environment could be achieve with smaller window displacements and may have actually *reduced* search time by increasing legibility.

This outcome highlights a tradeoff in the dual application of saliency and constancy in the Personal Façade; while our results clearly show that spatial constancy allows windows to be found efficiently, attention to background saliency may counter the benefits of constancy in environments with a high visual density. Although users will eventually learn the window positions in any regularly-visited environment, those with an abundance of salient regions may cause some windows to be more difficult to initially locate. One possible response to this finding would be to design future interfaces with multiple modes for different scenarios. For example, one mode would boost *Adherence* to minimize window search time. An alternate mode would boost *Non-occlusion* to minimize occlusion.

5.6 Summary and Future Work

Through this first exploration of the Personal Façade, we take away several lessons:

1) Spatial constancy is key to application switching efficiency in a limited FoV window layout interface.
2) Yet, it is possible to strike a balance between the conflicting constraints imposed by spatial constancy and visual saliency, with the impending tradeoffs in efficiency determined by the environment's complexity.
3) Environmental interference can be to some extent overlooked (e.g. placing a window on a partially oblique surface, occluding objects of lesser importance)

to favor purely spatial concerns (e.g. avoiding large head motions, close grouping of related windows).

We acknowledge that our contributions contain several limitations. First, we have explored only two environments of a great possible variety. Also, beyond application layouts, techniques both for initiating layout operations and for interacting with window content require investigation. We acknowledge these and other drawbacks as we outline several directions for future exploration:

Computational load: Our Personal Façade prototype requires greater hardware capabilities than are typical of current wearable displays. Computation efforts are used for reconstructing the environments, estimating saliency and optimizing layouts. Nevertheless, individual components of the implementation, such as popular visual saliency models, are highly amenable to optimization, for instance with parallel computing (e.g. on GPUs). In the case of this work, there also exist highly optimized lightweight FPGA solutions [14] that imply very fast operation and low power consumption. Furthermore, there exist alternative processing models, including directly leveraging RGB-D data and foregoing the need for a mesh model. Surfaces for displaying content might be detected during geometry recovery using techniques such as Dense Planar SLAM [181] or Parallel Tracking and Mapping [113].

Future wearables and smart environments: Forthcoming developments in both display and sensor technology increase the likelihood of a wearable system becoming capable of supporting our envisioned prototype. This includes significantly broader fields of view (e.g. [132]), and mobile and miniaturized depth based sensors (e.g. Google Tango, Occipital, Microsoft HoloLens) especially suitable for portable and wearable applications. It may also be expected that our future everyday surroundings may be equipped with sensors that detect the environment's structure, internal motion and additive saliency. These environmental data could then be accessed by multiple client wearables to save device load. However, these forthcoming improvements bear implications toward our findings; in particular our findings on the importance of spatial constancy, may not generalize to future devices with wide FoVs that allow increased use of peripheral vision.

Temporal considerations: Unlike our test environments, real-world environs are not static. For example, lighting conditions may change throughout the day, surfaces such as window blinds often move frequently and there may be people moving to and fro. These issues present additional design problems; for instance, if a passerby enters a scene, does the user prefer the window to be temporarily occluded, or to dynamically shift out of the way to remain visible? We may easily adjust our system to include additional rules, for instance to regard as salient any region where salient objects regularly appear during long-term sensing. Additional detectors, such as specific objects detectors, will allow complex semantic rules. For example, placing a clock application above a room's door may carry semantic inferences that some users are accustomed to.

5.7 Conclusion

We introduce the Personal Façade, a HWD interface that integrates application windows into the built environment. Our implementation of the Personal Façade focuses on blending the principles of spatial constancy and visual saliency into a spatial window management interface. We implement an algorithm that applies these and other constraints to produce window layouts that we demonstrate in two test environments with varying visual information density. We run a user study to show the effects of different combinations of constancy and saliency in these environments. In summary, we successfully demonstrate layouts that provide efficient application search while also observing physical differences between user environments.

6 Natural and Effective Interaction

This chapter focuses on the SAI requirement of **interactivity**. Previous chapters explored questions about how to layout information for effective consumption. To support meaningful work with analytic tasks, users must be able to dig in and explore information, not just consume what is presented. Important actions that bring the human into the information sensemaking process, such as posing queries, making comparisons and scaling data visualizations, depend on the user's ability to interact effectively with the system.

There are many possible options for interacting with HWD content; however, there is as yet no standard method that has been proven effective for spatial interfaces the way the mouse and keyboard are for desktop use, or touch gestures are for mobile devices. One practical benefit of spatial user interfaces is their support of *naturalism* [32], meaning users interacting with virtual content behave in a manner very similar to everyday interaction with real-world objects. Researchers have shown many benefits of naturalism in 3D environments [32]. For instance, it is highly intuitive to select an object with a pointing gesture, or to move and rotate an object using direct manipulation. However, these natural techniques are prone to fatigue from excessive arm motions [58,91] and limited precision due to the absence of haptic feedback [199].

In designing techniques for SAI interaction, we can borrow from considerable research done on 3D spatial interaction techniques in the VR realm. However, these techniques often rely on external tracking systems [30,165], bulky input devices [29] or haptic surfaces [199]. In contrast, a practical interface for interaction with SAIs must consider new requirements not always relevant in VR labs.

For instance, we have already discussed the importance of mobility for SAIs to allow tasks to be performed in-situ, at the convenience of the user, rather than as determined by constraints of the technology. This requires input devices that are small enough to either wear or carry in a user's pocket, without occupying a user's hands or otherwise interfering with normal activities. Another important aspect is social acceptability. User interactions in many instances should be minute enough to remain subtle, thus avoiding unwanted attention in public spaces [4,11] and preventing user fatigue from substantial movements of the arm, neck or eyes [91]. Although an input device should be small and inconspicuous enough to blend into the user's typical attire, the device should, at the same time, provide cues to make others aware when the user is engaged with the interactive system [57]. Overly subtle interactions that lack social cues can be problematic; for instance, when interaction interrupts a conversation, the user's attention should clearly indicate whether it is directed at the other person or at the device.

In this chapter we propose a new method of interaction with SAIs by combining a wearable input device with computer-vision-based hand tracking (**Figure 31**) [55]. In particular, we adopt a ring form factor, which has recently gained attention [10,41,110,157,228], for example in providing subtle [6,29], always-available input [10,41,228] or gesture commands [10,41,228]. Meanwhile, miniaturized depth cameras [76] mounted on HWDs have recently been used to track a user's hands [80,125,131,146,204],

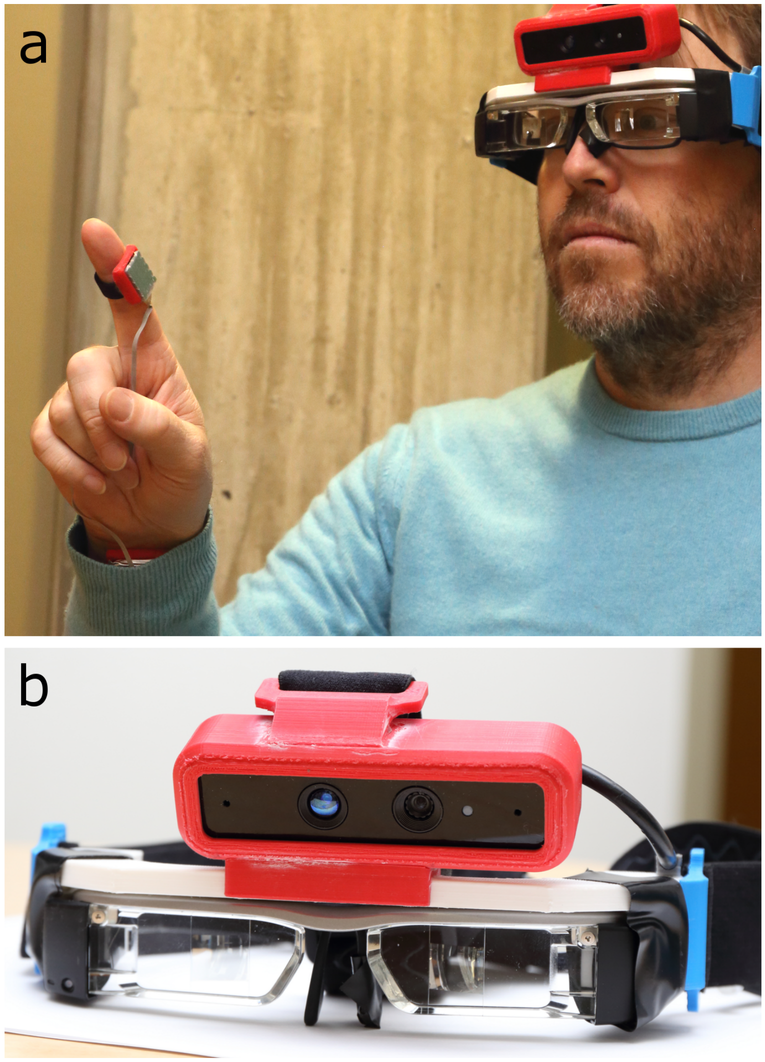

Figure 31: HWD and Ring Device Apparatus
a) We propose a novel combination of computer vision-based hand tracking and ring device input, to provide direct manipulation with precision and low fatigue. b) Miniaturized depth cameras allow natural hand gestures, such as grasping and pointing, to be used for input on HWDs.

providing natural direct input techniques. Our work is the first to explore hand tracking in conjunction with ring input, to leverage the combined benefits of these techniques.

Our motivation for exploring this combination of technologies is to support effective interaction for in-situ analytic tasks. Primarily, we want to provide an interface that supports naturalism, but also has the high precision necessary for selection of items in detailed visualizations. HWD-based hand tracking supports naturalism, but alone the technology is not sophisticated enough to provide the needed precision. A ring interface is small enough to not interfere with natural pointing and grasping gestures, while providing a high-precision interface. Also, this interface is literally at the user's fingertips, eliminating the costly time needed to acquire a device. With this interface we further aim to support low fatigue interaction, mobility and social acceptability. The input device and hand tracking sensors are embedded in devices worn on the user's hands and face, eliminating the need to carry additional heavy or awkward hardware. Given time for refinement, these wearable devices will attain form factors with very small size and low weight, and will be difficult to differentiate from jewellery or eyeglasses. Such devices will be readily mobile, and won't instil fatigue, or interfere with a user's everyday physical or social interactions with the outside world.

One final but important concept we present in this work is support for *dual-tier interaction*: the ability to interact with SAIs at both the *layout* level and the *content* level. Effective analytic taskwork requires interaction at both levels, ideally supported by a method that allows each to be done seamlessly, without awkward mode-switching requirements costing time and cognitive workload. We propose that the combined use of ring and hand tracking input can support seamless dual-tier interaction by combining periodic natural gestures at the layout level, with low-fatigue, precision input for intensive work at the content level. We implement a prototype interface and provide several proof-of-concept examples of how this setup can be used to support effective interaction with SAIs.

6.1 Related Work

Natural 3D Interaction Techniques

Considerable work has been devoted to creating 3D input techniques in the VR realm. While some early research carried forward the cursor metaphor from desktop interfaces [234], later work realized the benefits of 'natural' motions such as grasping and pointing. In many cases, natural interactions have been shown to outperform abstract techniques that ignore our inherent understanding of motion in 3D space [32].

One category of techniques, *virtual hand,* is so named because it employs a simulated hand to mimic the motions of a user's real hand. A benefit of virtual worlds, however, is that the constraints of the physical world may be overcome, for instance by stretching the virtual hand beyond the user's natural reach [30,165]. Such 'hyper-natural' techniques [32] leverage naturalism to reach beyond it to provide additional benefits possible only with virtual objects that are not bound by natural laws.

A second category of techniques, *raycasting* [144], extends the user's pointing limb with a virtual pointer, often giving the appearance of a laser beam. Raycasting techniques are widely applied in 3D virtual environments, but are not without limitations, such as selecting small objects from a distance and selecting objects that are occluded from view. Accuracy is improved by widening the ray into a cone [122], but this requires a secondary technique to disambiguate individual items from clusters [9]. Similar methods have been proposed for disambiguating items that lie along the path of a ray [211], as well as alternative techniques such as bending the ray [158]. For further details about the wide variety of 3D interaction techniques developed for virtual environments, we refer readers to an excellent survey by Argelaguet and Andujar [9].

With a few exceptions [80,118,162], natural interaction techniques have not been widely studied in AR environments. Our work aims to supplement these existing techniques with input from a ring device to provide a sense of naturalism while countering the drawbacks of direct interaction with virtual objects, such as low precision [124,199] and fatigue [91,94].

Hybrid Interaction Techniques

Another body of related research is in the area of 'hybrid' interaction techniques. A significant portion of this work has resulted from relatively recent interest in interaction with large, or wall-sized, displays. One particular research question with such displays is how to interact efficiently with a large display area. Conventional display interaction broadly falls into two categories: *Direct* input, such as touch or stylus input, and *indirect* input, such as a cursor driven by a mouse or trackpad. Direct methods generally use an *absolute* mapping of input space to display space, whereas indirect methods typically use *relative* mapping. These categories have different areas of strength and weakness, for instance the utility of indirect methods, such as a mouse on a desktop or a trackpad on a laptop, versus the intuitive use of direct input, such as a stylus for writing on a tablet or the swipe of a finger on a smartphone.

None of these interaction methods, when used alone, adapts well to large displays. Different levels of scale are required for different actions, such as viewing a large region to gain a holistic context, or focusing in on a small region for precise selection. For example, direct touch requires walking to distant targets and forces users to view distant regions from an oblique angle [151]. Conversely, indirect methods can be used while viewing the entire display from afar; however, moving a mouse cursor a long distance requires many inefficient and fatiguing clutching motions [136]. For indirect input, there is no single scale that allows both fast cursor motions to reach distant targets, and precise motions for selecting them. One method for overcoming this problem lies in dynamic scaling of the input-display mapping, known as *cursor acceleration* [68]. However the correct tuning of this function is a complex problem [151].

Several effective methods rely on a mixture of selection techniques [67,136,151,216] to support the contrasting operations of traversal and precision. For instance, the ARC-Pad technique of McCallum et al. [136] uses absolute pointing on a handheld device to select a corresponding region of the display, followed by indirect, relative motion to acquire a target. A technique proposed by Nancel et al. [151], on the other hand, uses head motion mapped indirectly to display space for coarse selection, followed by direct hand motion for precision. The trade-off of these techniques for the precision gained is they require switching between multiple modes. Thus primary focus of hybrid techniques is the reduction of switching costs between modes so the cost of switching between techniques does not outweigh their combined benefits.

Aside from combining direct and indirect, or relative and absolute techniques, researchers have also explored 'multimodal' techniques that combine multiple methods of natural interaction [118,162]. Piumsomboon et al. [162], for example, combine hand gestures for manipulating virtual objects seen through a HWD, with voice commands for mode selection. Similar multimodal interfaces that combine direct manipulation with voice commands have been adopted for soon-available commercial devices such as Meta Glass [138] and Microsoft HoloLens [139]. Like other hybrid techniques, the mode-switching required by these multimodal techniques costs time and increases cognitive load.

In contrast to the techniques outlined here, we employ direct input through grasping and pointing gestures, with indirect input on a ring device, a form factor we discuss next.

Ring-Based Input Devices

Early work by IBM produced a ring-based device with a single button for user input [145]. Other early devices used magnets sensed by external magnetic tracking to avoid bulky batteries and wires [10,85]. Miniaturization has recently led to a number of more sophisticated ring devices used for a variety of purposes including 1D rotational input [10], gesture recognition [111,223], and discreet-touch input [41]. Ogata et al. [157] added small IR sensors to the interior of their device, iRing, to detect changes in skin reflectivity. They demonstrate that pressure from bending the finger or applied by external objects can be detected with reasonable accuracy and mapped to various commands such as a music player. Chan et al. [41] created FingerPad by mounting a magnetic Hall sensor to the user's thumb nail. The thumb position is read by a sensor on the opposing index fingernail as the user traces small gestures between the two digits, such as digits of a password or swipe gestures. Magic Finger from, Yang et al. [228], uses a mouse's optical flow sensor to trace a finger's motion on a surface. Similarly, Kienzle and Hinckley [110] instrument the finger with an infrared sensor and a single axis gyroscope to allow recognition of traced gestures. Yet another device, called Plex [232], follows a tradition of glove-based devices [194], but consists of a textile covering only a single finger. This device allows swiping gestures produced by the user's thumb against the outer edge of the index finger, and among the devices discussed here is the most similar to the one used in our implementation. However, ours is the first work, to our knowledge, that combines a ring device to supplement natural input produced by a wearable depth camera.

Describing Hybrid Techniques

The hybrid techniques discussed above may be described according to various dimensions of a design framework [58]. Potential dimensions might include the number of input devices, and the nature of the different modalities (e.g. absolute vs. relative, direct vs. indirect, voice vs. hand). Another useful dimension to explore is the temporal relationship between modalities, as described in the framework of Vernier and Nigay [212]. Five potential relationships they described are shown in **Figure 32**. We use similar diagrams in the figures below to help depict the relationships of the various modalities used in our prototype implementations. Our goal is to show how a wide variety of rich interaction techniques can be created by combining ring device input with hand-tracking on a HWD. The techniques we demonstrate are aimed at providing natural input for use with SAIs, while improving precision and minimizing fatigue.

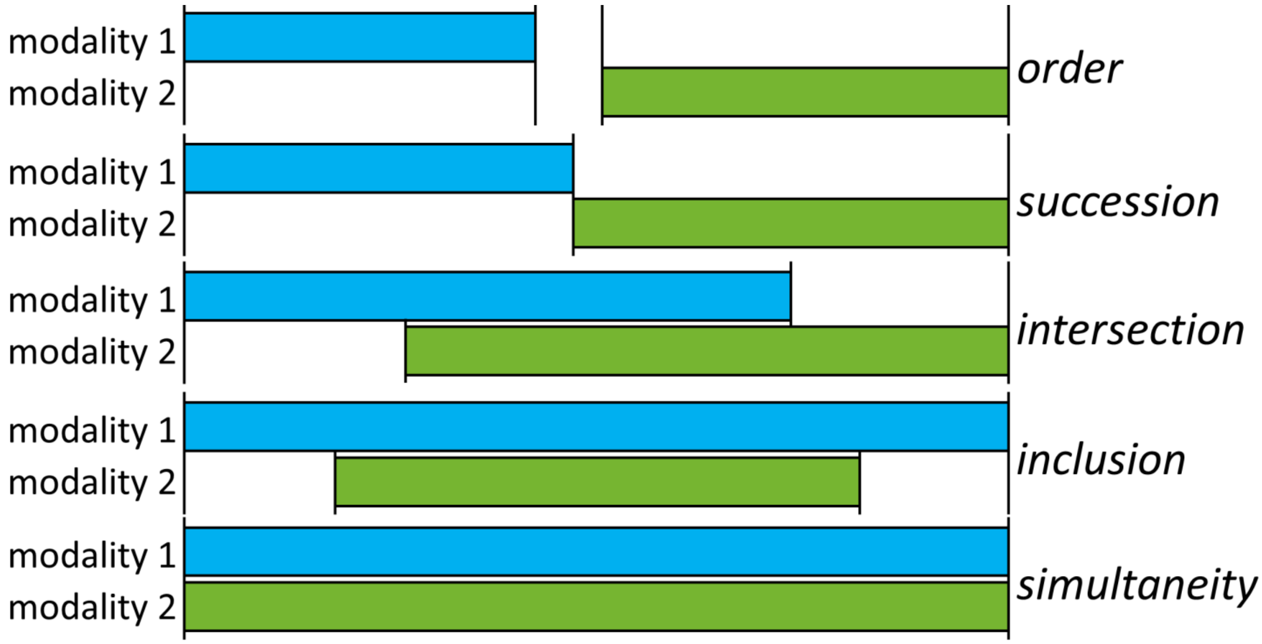

Figure 32. Temporal Relationships of Hybrid Techniques
We apply the framework of Vernier and Nigay [212]**, which characterizes various ways to combine input modalities, to describe different potential mixtures of ring and hand-tracking input in our prototype implementations.**

6.2 Dual-Tier Interaction

As discussed in our initial list of requirements for SAI, analytic tasks require users to explore one or more data sources to learn information and make decisions. The specific type of interaction required depends on the nature of the task and content. However, most workflows can be abstracted to a small set of basic operations. In 3D user interfaces, the primary operations are commonly referred to as the 'big five' tasks of navigation, selection, manipulation, system control and symbolic input [29]. In this work, we are concerned primarily with the selection and manipulation of objects.

In the Ethereal Planes metaphor, these operations can be applied to two different tiers of the interface: One for managing the layout of 2D views in the surrounding 3D space, and the second for interacting with the content that resides within the 2D views. **Table 8** lists a number of common operations and provides examples showing how each of these can potentially be applied to either the layout tier or the content tier. The following subsections discuss considerations for each of these tiers. We also include an overview of various layout operations we explored during our work on the Personal Façade interface we introduced in Chapter 5.

Operation	**Layout**	**Content**
select	choose window in focus	highlight one or more items
move	translate or rotate windows in 3D space	pan content to bring items into view
resize	make a window larger or smaller	zoom in or out to change scale of items
change	open or close a visualization	change the representation of a chosen view
filter	choose which views are relevant	reduce the amount of content shown in a view
symbolic input	invoke system or menu commands	text entry, numeric input, sketching

Table 8. Dual-Tier Interaction Examples
Operations for interacting with virtual 2D views in 3D space must consider interaction on two 'tiers'. Coarse gestures may be used to manipulate the layout of multiple views, while fine-grained interaction is required for selection and manipulation of visualization content.

Interaction with Window Layouts

In previous chapters, we primarily explored automatic window layouts, with the assumption that a pre-defined configuration exists for a user's particular situation. However, there are many instances where users may want to manipulate window layouts themselves, for instance to define the default configuration used by the Personal Façade's automatic layout generator. Window layouts can also be used to assist in analytic tasks. For example, users may want to place two linked data visualizations side-by-side for comparison, or to carefully overlay multiple layers of a map.

We envision that natural input techniques may be used within SAIs [59] in a manner analogous to the way touchscreens are used on modern laptop computers, alongside mouse and keyboard input [44]. Direct manipulation is used infrequently for fast and intuitive control of coarse-grained objects (e.g. swiping in the control panel or closing an application by dragging it to the bottom of the screen), while indirect techniques support precision and long-term use. Similarly, direct manipulation could support the manual arrangement of windows within a layout.

In the Personal Façade, we implemented several manual operations to give users control over individual windows and window relationships. Below we describe several such operations that we implemented. This exploration is limited to the operations themselves and does not give broad consideration the optimal interaction techniques for invoking them. We use a simple ray-casting medium with a handheld wand and the wand's embedded buttons for invoking commands.

Moving and Resizing windows

Core to the usability of all window layout managers is the ability to manually rearrange content. In our implementation, users can reposition a window by selecting it with the wand and then pointing to a new location (**Figure 33**a). Users can similarly resize a selected window with the wand buttons. If a collision is detected given the new configuration, then other windows are locally repositioned while the moved/resized window is 'pinned' in place.

Stitching and piling

Additional operations involve relations between multiple windows. Users can select two or more windows and stitch them together; selected windows move alongside the target window resize along the adjoining seam. Likewise, windows can be piled on top of a target window (**Figure 33**d). The stitched/piled windows can subsequently be repositioned as a single object.

Saving and Restoring Configurations

Once a configuration has been manually created, users can save this configuration in the form of a body-centric array [22,56] for mobile use (**Figure 33**b). The configuration can later be integrated into a new environment (**Figure 33**c). Furthermore, the user can choose the appropriate layout mode (*Balanced, Constancy, or Saliency*) to suit the situation.

Orthogonality adjustment

Windows in the Personal Façade can potentially be placed on awkwardly-aligned surfaces that affect legibility, such as along a long hallway or on desktops. For such situations, we implemented an operation that corrects a window's orientation, making it co-planar to the user's FoV. Selecting a window and holding the wand trigger makes a window 'stand up'. It realigns with the surface when the trigger is let go.

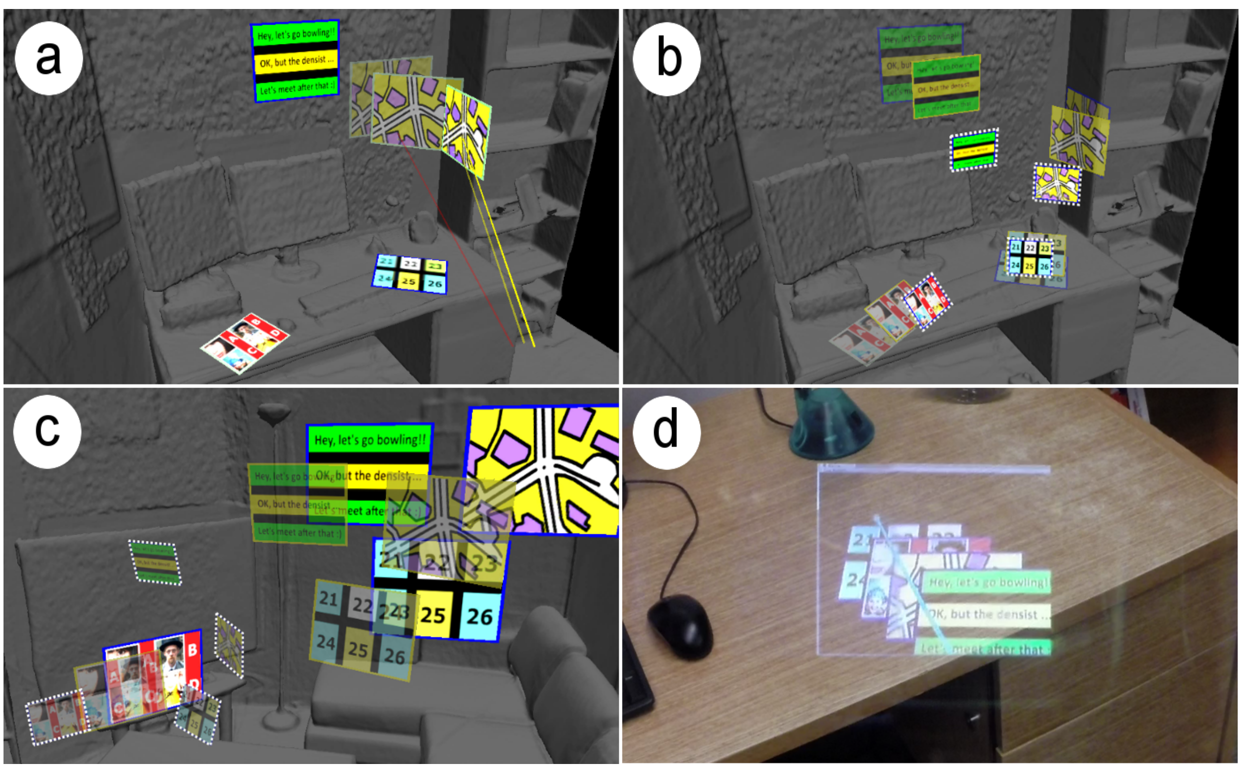

Figure 33. Personal Façade Interactions
a) Users can reposition windows to create custom configurations. These can be saved to body-centric format (b) and re-applied in new environments (c). We implemented several other operations such as piling (d).

'Space-saving' windows

In consideration of highly cluttered environments, we can set the Personal Façade to a space-saving mode, which minimizes all windows to thumbnails. Windows remain in this form until the user holds her gaze in a window's vicinity, at which point the thumbnail grows to the full-size of the user's display. It returns to normal size once the user's gaze is diverted.

Help with finding windows

The environment's structure can significantly impact the layout of windows, particularly in Saliency-based layouts. To assist with recall of windows in unfamiliar environments, we implemented visual cues that link each window's current position to a transparent proxy in its body-centric, default position. This allows users to familiarize themselves with a new layout, or to re-formulate their mental model in new surroundings.

While other operations on the windows are possible, such as adding a physics engine to 'bump' windows and move them around as in Bumptop [1], we settled on these basic operations as they demonstrate the ability to deviate from an automated layout to provide the user with fuller control.

Interaction with Window Contents

Items within a window are by nature much smaller than the window itself, so interactions at the content level will require a higher level of precision. For object manipulations, unless items are to be moved between windows, the amplitude of motion will be relatively small compared to the window level. Also, whereas moving windows at the layout level is expected to occur relatively infrequently, interaction at the content level will likely occupy a significant portion of the time spent interacting with a SAI.

At the content level, it is also important to consider the SAI requirement for interpretation of information. Furthermore, the interpretation and interaction components are highly interdependent. For instance, during our user studies on the personal cockpit, we learned that users completed our task more quickly when the interface was fully within the FoV. From this we can assume that it is a best-practice for SAIs to make individual application windows self-contained, such that all of the information required for a particular task is present on a single interface. However, information is sometimes easier to interpret when spread between multiple, simplified visualizations [15,192]. Also, these windows may be interlinked such that manipulating content on one window may affect the information displayed on other windows. The FoV limitations of HWDs impose further challenges on application developers when determining the content layout on a set of SAI windows.

A Dual-Tier Interaction Scenario

To demonstrate the technical challenges that dual-tier interaction poses in a SAI, let us take the example analytic task of a hotel search. Imagine a traveller is performing this task in-situ on arrival in a new city. One possible SAI design is to spread the required information and interactive elements among three windows as shown in **Figure 34**. In the centre window is a map showing the location of several hotels. To the left is a preview window that provides detailed information about the hotels, and to the right is a filter control panel, where users can specify desired attributes (in this case, cost, star-rating and guest review scores). The windows are interlinked [231] such that selecting a specific hotel on the map causes details about that hotel to appear on the preview pane. Likewise, adjusting the filter controls reduces or increases the number of hotels shown on the map.

This scenario requires specific operations of item selection (selecting a hotel on the map or selecting a slider) and object manipulation (moving a slider to the left or right). Additional operations might be included, for instance zooming to reveal smaller regions on the map, scrolling through a long list of hotel details, or navigation through a greater selection of filter parameters. There are numerous ways these operations could be accomplished, using a variety of different interaction techniques and input devices. For instance, one of the sliders could be selected with a relatively coarse gesture that could be easily detected by a hand-tracking camera. However, selecting one of the small hotel pins on the map will require a greater level of precision. The ability to precisely select items will require accurate and robust hardware; however, travellers looking for a hotel may appreciate the ability to operate this SAI without carrying a bulky input device. Such trade-offs of direct and indirect input methods are discussed in the following section, followed by our example implementation of this hotel search application.

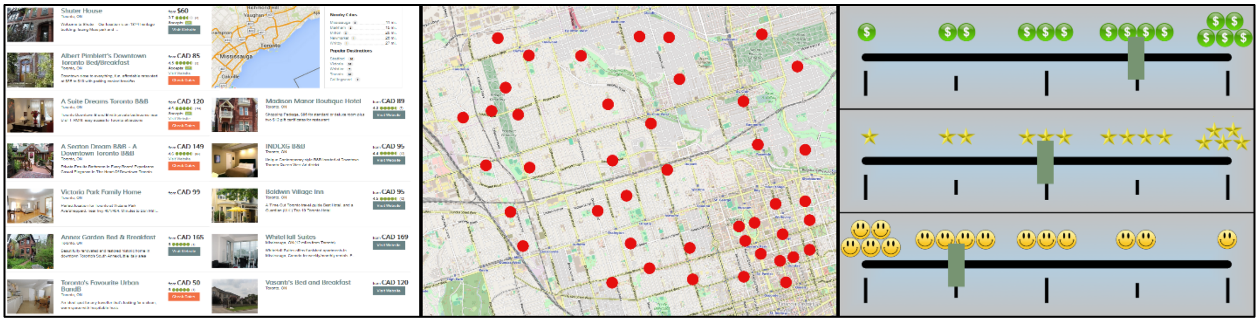

Figure 34. Demonstrated Analytic Task
A hotel search is an example of an everyday analytic task that would require content-level interaction within a SAI. One potential implementation would use three windows, a hotel preview window (left), a map (centre) and a filter panel (right). Manipulating the filters reduces or increases the number of hotel 'pins' on the map, and selecting different map pins reveals details about those locations in the preview pane.

6.3 Direct and Indirect Input Methods for SAIs

For interacting with SAIs, there are a number of potential interaction technologies that support either direct or indirect input methods. In this work, we consider methods that use small input devices, or hand tracking methods that do not require a device (**Figure 35**), to allow mobility for in-situ tasks with minimal interference in normal activities. Two options of such viable methods are camera-based hand-tracking for direct input, and a ring-based sensing device for indirect input. Hand tracking supports direct manipulation using the hands (**Figure 31**a) or raycasting using a finger or small pointing device (**Figure 35**a). A ring device can be highly versatile using only a few widely-available sensors, to allow simple input techniques, such as tapping, 1D twisting (**Figure 35**b), 2D swiping (**Figure 35**c), or 3D rotation. Advantages of these methods are outlined in **Table 9**.

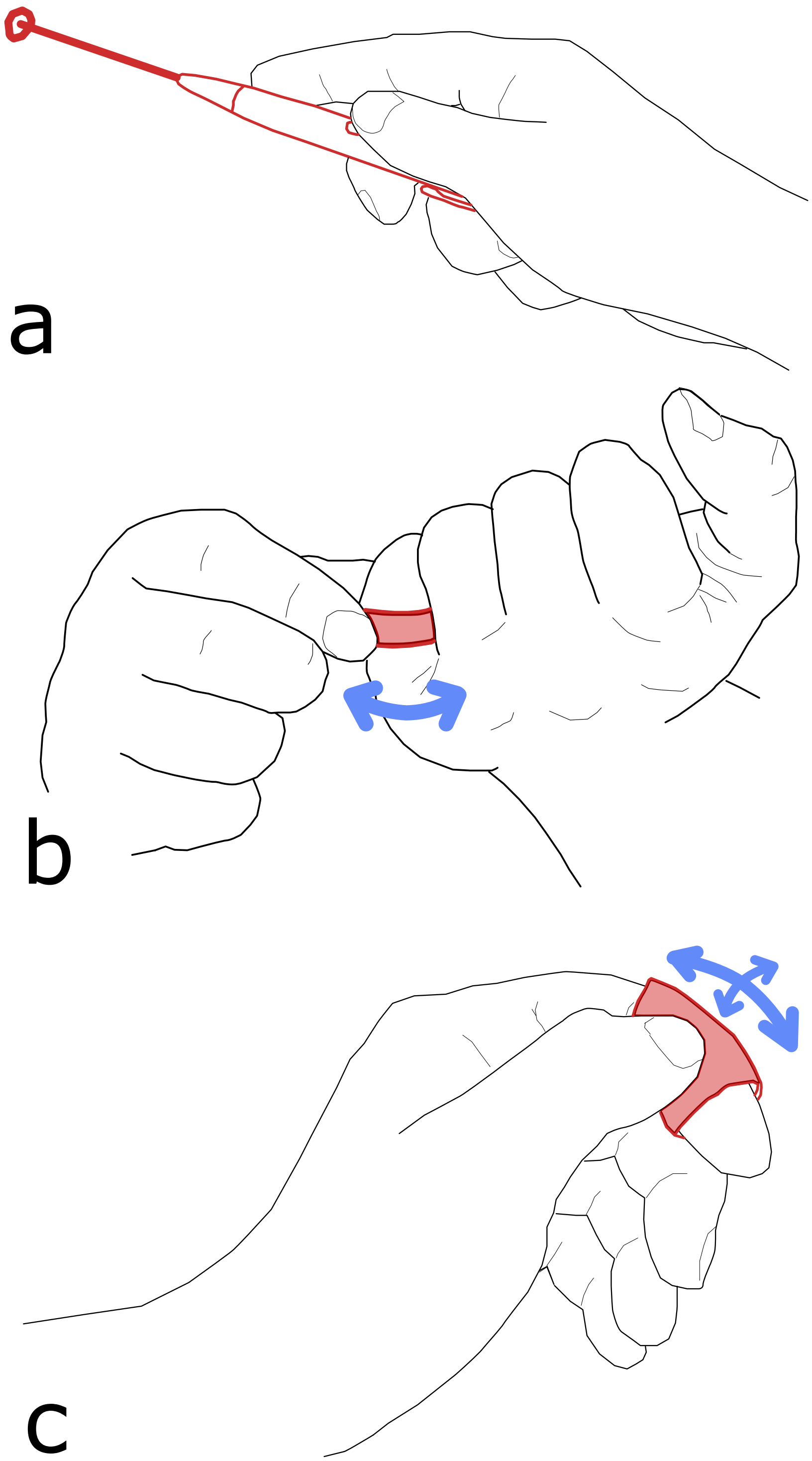

Figure 35. Examples of Lightweight Interaction Apparatuses
A variety of handheld or wearable devices can potentially be developed to provide interaction with SAIs. Possible form factors include a stylus for pointing (a), a ring for scrolling (b) or a finger pad for 2D input (c).

Using one's hands is likely the most natural and intuitive way to interact with a 3D interface [32,94]. Hand tracking allows direct, absolute selection of objects via grasping motions, and direct manipulation of objects in 3D space using arm and wrist motions. One limitation of the hands is that the user can only interact in such a way if objects are easily within reach, both within range and located in a position that provides ergonomic access. Also, precision with hands motions is limited by several factors, such as finger size and hand occlusion. Wearable sensor technologies are prone to noise and may not be capable of precise tracking with a high level of reliability in all environments. Furthermore, users may have difficulty performing very precise gestures without a haptic surface for feedback [199]. One further drawback of hand input is fatigue, likely to result from large motions over an extended duration without an appropriate resting surface.

Pointing is another natural action often used in everyday activities. Pointing gestures may be made using a finger (usually the index finger) or with an object, such as a stylus. In 3D interfaces, the pointing member is often extended with a visible ray, in a technique known as raycasting [143]. Raycasting allows selection of distant objects; however, effects of hand tremor reduce precision over distance. Extending the pointing limb for long periods may also induce fatigue.

A ring device with fully-contained sensing and networking components can provide a variety of simple interaction techniques without the need to carry a bulky input device. If sufficiently small, such a device is highly mobile and will only nominally interfere with natural, real-world activities. The supported interaction techniques do not provide

	Hand Tracking (direct input)	Ring Device (indirect input)
Naturalism	yes	no
High Precision	limited	yes
Low Fatigue	short durations	yes
Mobility	yes	yes
Low Encumbrance	none	minimal

Table 9. Advantages of Hand-Tracking and Ring Device Input
Hand tracking and ring input have different weaknesses, and several complementary benefits. Hand tracking supports naturalism, but has limited precision and is susceptible to fatigue. Ring input is abstract (no naturalism), but allows higher precision and lower fatigue. Both can be combined to provide the coarse and fine interactions required for dual-tier interaction with spatial layouts.

naturalism; however, the device can potentially support high-precision interaction techniques, and can be used for extended periods without fatigue.

6.4 Implementation

To explore interactions that combine direct and indirect input, we created a prototype implementation that sends ring device input and hand tracking data to a HWD. For hand tracking, we mounted a Softkinetic DS-325 depth camera in a 3D-printed housing unit on top of a Moverio BT-200 HWD (**Figure 31**). The camera is tethered to a desktop computer, which processes the data using SoftKinetic's iisu middleware and sends the processed data to the Moverio over a TCP connection. These data include the hand centroid position, finger and thumb positions, pointing tip position, and hand state (open/closed).

To supplement direct hand input, we use a small ring device (**Figure 36**) capable of basic and well-known operations, such as tapping and flicking. The device was designed by Ahmad Byagowi, a PhD Candidate in the University of Manitoba Faculty of Engineering. The device contains a small capacitive touch sensor and a nine-axis inertial measurement unit (IMU). The capacitive touch sensor is composed of an array of capacitors arranged in a 3 × 4 grid on a surface measuring 12×16 mm. The capacitors are connected to a Microchip MTCH6102 controller, which sends position and gesture data with a resolution of 384 × 567. A Bosch BNO055 IMU module contains an Atmel ARM Cortex-M0 processor, and provides absolute pitch, roll and yaw. Both the touch control and IMU are interfaced using an I^2C bus, requiring only four wires for connection.

To fit comfortably between two joints of an average human finger (**Figure 36**d), the components are divided between two stacked boards. One contains the capacitive sensor grid (**Figure 36**b), and the other contains the two controllers (**Figure 36**c). Sandwiched between the two boards is a sheet of copper shielding.

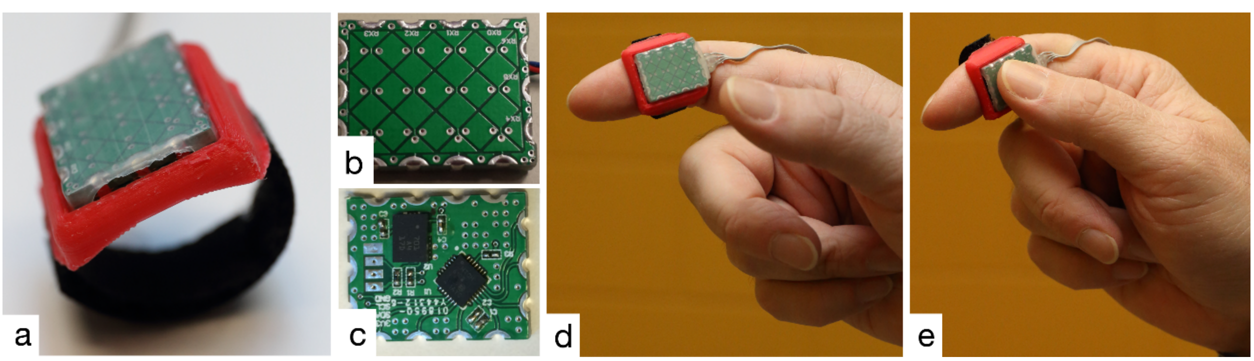

Figure 36: Ring Device Apparatus
Our ring device (a) contains a miniature trackpad composed of an array of capacitive sensors (b) and a nine-axis IMU (c). When worn on the index finger (d), this small device supports simple interaction techniques such as tapping (e) and swiping.

The unit is attached to a 3D-printed base, and affixed to the wearer's finger by a hook-and-loop fastener strip (**Figure 36**d). Data are relayed to the HWD via Bluetooth through a tethered Arduino microcontroller. Data filtering and all other processing are done on the Moverio unit, which runs Android 4.0. We developed the HWD program in Unity 3D.

6.5 Implemented SAI Interactions

We implemented several interaction techniques that leverage the benefits of direct and indirect input. Our intention is not to create new 3D interaction techniques but to demonstrate the implementation of existing techniques drawn from the great body of existing literature. We show how these techniques can be supported in a wearable form factor to allow effective interaction with SAIs. We demonstrate these implementations using the novel combination of hand tracking with a ring device, as described in the previous sections.

6 DoF Direct Manipulation

Given a depth channel, computer vision algorithms are now capable of robustly segmenting human hands and tracking the positions of centroids and fingertips. However, accurately measuring the absolute rotation of hands is more challenging, particularly if the hand is closed or fingers are otherwise occluded. Conversely, IMUs are now capable of providing very accurate absolute orientation but cannot accurately track position. By combining the ring's IMU with the HWD's depth camera, we can enable tracking in six degrees of freedom (DoF).

In our implementation, this sensor combination allows windows to be freely manipulated in 3D space (**Figure 37**, **Figure 38**). After grabbing a window, by closing a hand around its virtual bezel, the user can position the window anywhere within reach.

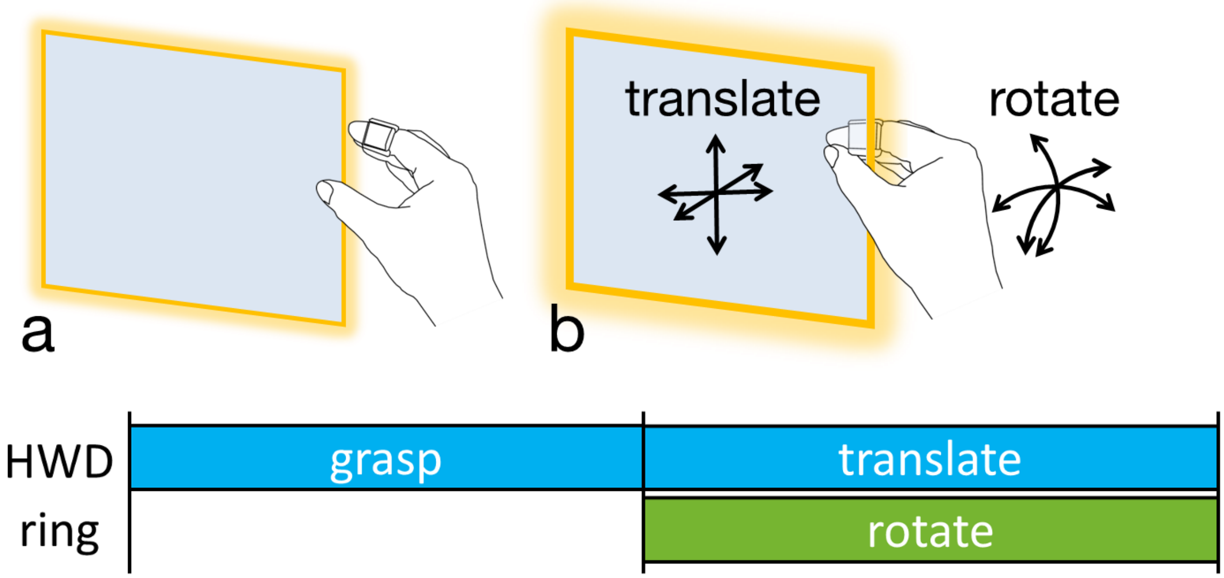

Figure 37. SAI Direct Manipulation
Direct manipulation is useful for infrequent, coarse-grained gestures. In this figure, a user repositions a virtual window using direct manipulation. a) A user 'grabs' a window using a grasp gesture. b) Using combined data from the HWD-mounted depth camera and ring-embedded IMU, the window can be translated and rotated freely in 6 DoF.

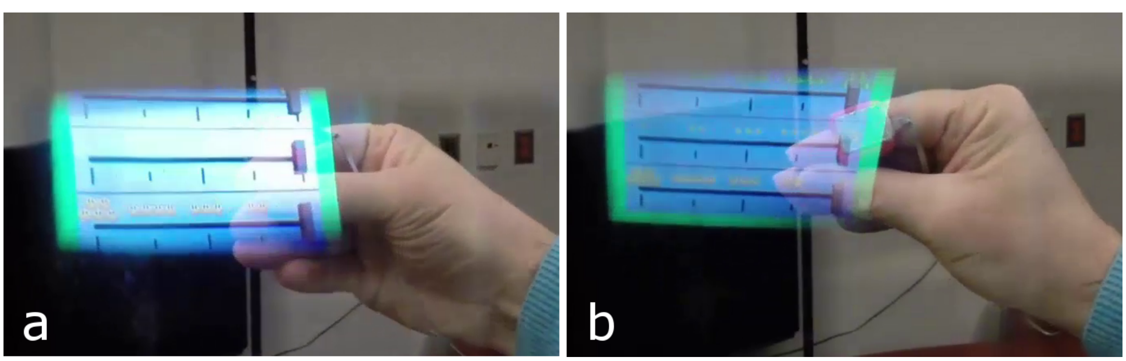

Figure 38. SAI Implementation: Direct Manipulation
Our implementation, as viewed through the lens of the HWD. The combined input devices allow direct manipulation of a window in 6 DoF.

Window Selection

During an analytic task, users may wish to select a particular window to apply subsequent operations (e.g. zoom) or to interact with its contents. In our implemented system, a window may be selected using two variations of natural input methods.

The first method uses direct manipulation; the user simply 'taps' the window's virtual surface with an extended finger (**Figure 39**a, **Figure 41**a). The second method uses raycasting [143], a common method that projects a virtual laser beam from the user's pointing finger or instrument to provide feedback and assist precise pointing motions. A tap on the ring enables a ray that extends from the user's hand. A second tap selects a window (**Figure 40**a, **Figure 41**b), if found, or disables the ray. With either technique, the closest object to the given point is selected for further use (**Figure 39**b, **Figure 40**b).

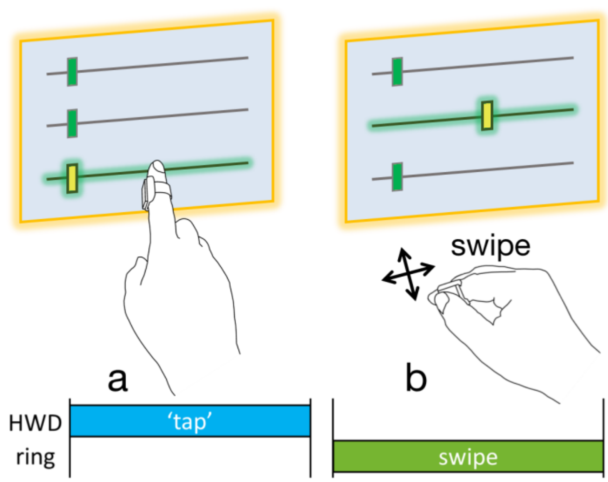

Figure 39: SAI Coarse Selection
a) Selecting a virtual window with a 'tap' gesture also selects the nearest control within the window. b) Swiping gestures on the ring pad may be used to change the control selection and change the value of the selected control.

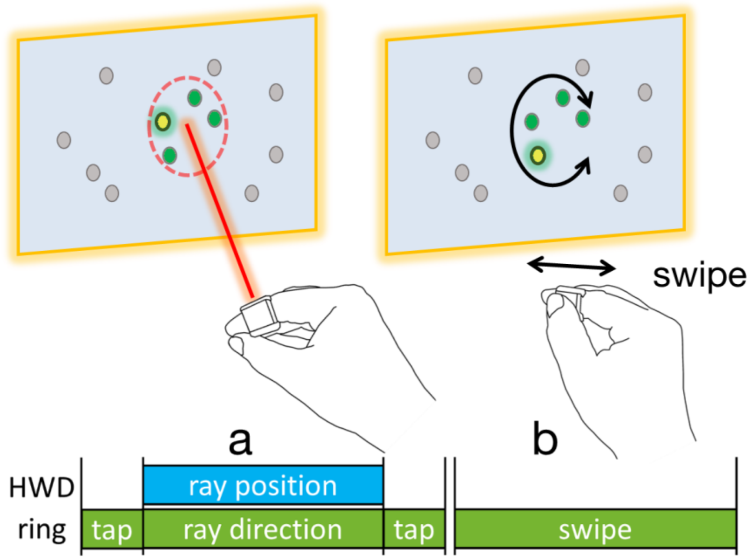

Figure 40: SAI Fine Selection
a) Items can also be selected using raycasting. Data points beyond a threshold distance from the ray selection point are disabled. b) Swiping the ring pad cycles through the enabled data points.

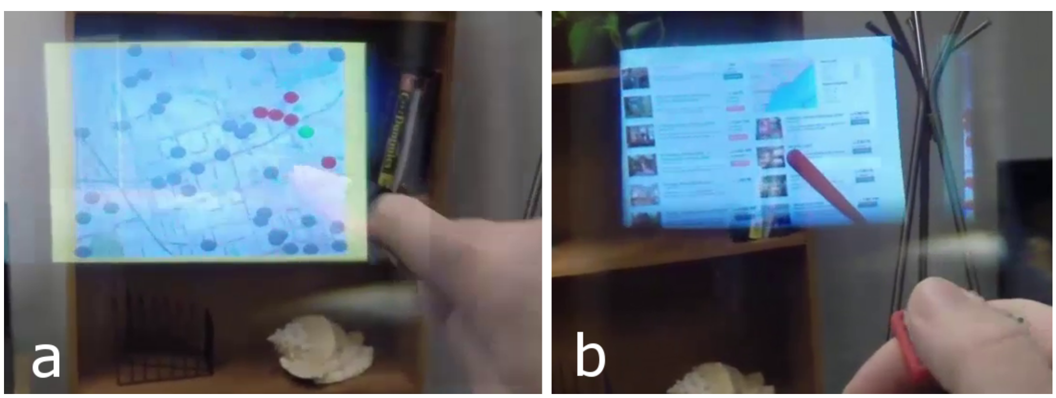

Figure 41. SAI Implementation: Direct Input Selection
Our implementation uses the combined HWD and ring sensors to provide different natural input methods. Users can select a window using a direct 'tap' gesture (a) on the virtual window surface, or by pointing a ray and tapping the ring pad (b). (The images in this figure and the ones below were captured directly through the lens of the Moverio Bt-200 HWD.)

Ray-Grabbing

When 6 DoF manipulation is not ideal, the ring's IMU allows more traditional 3D UI techniques [30,165]. For example, a user can 'ray-grab' [30] a window with a tap-and-hold gesture on the ring (**Figure 42**a, **Figure 43**a). The grabbed window can then be repositioned by mapping ring rotation to window translation in 2 DoF on a body-centric sphere (**Figure 42**b, **Figure 43**b). Lifting the thumb from the ring pad releases the window. The window then can be shifted in depth [165] by swiping up or down on the ring pad (**Figure 42**c, **Figure 43**c).

Control Activation and Small Object Selection

Hand position tracking is capable of allowing the operation of virtual controls on a 2D interface; however, extended reliance on direct manipulation can quickly cause arm fatigue [32,91]. Our system supports a combination of direct and indirect interaction methods. For instance, after selecting a slider control panel, the user can cycle through the vertically-aligned sliders by swiping up or down on the ring pad. The selected slider can then be moved by swiping left or right (**Figure 39**b).

Analytic tasks may also require the selection of data points on dense visualizations. Even assuming that current methods allow sufficient precision, research has shown that input precision suffers when a haptic surface is not available [199]. After making a coarse selection, we disable any points outside a defined threshold and allow users to refine their selection by cycling through the remaining points (**Figure 40**b).

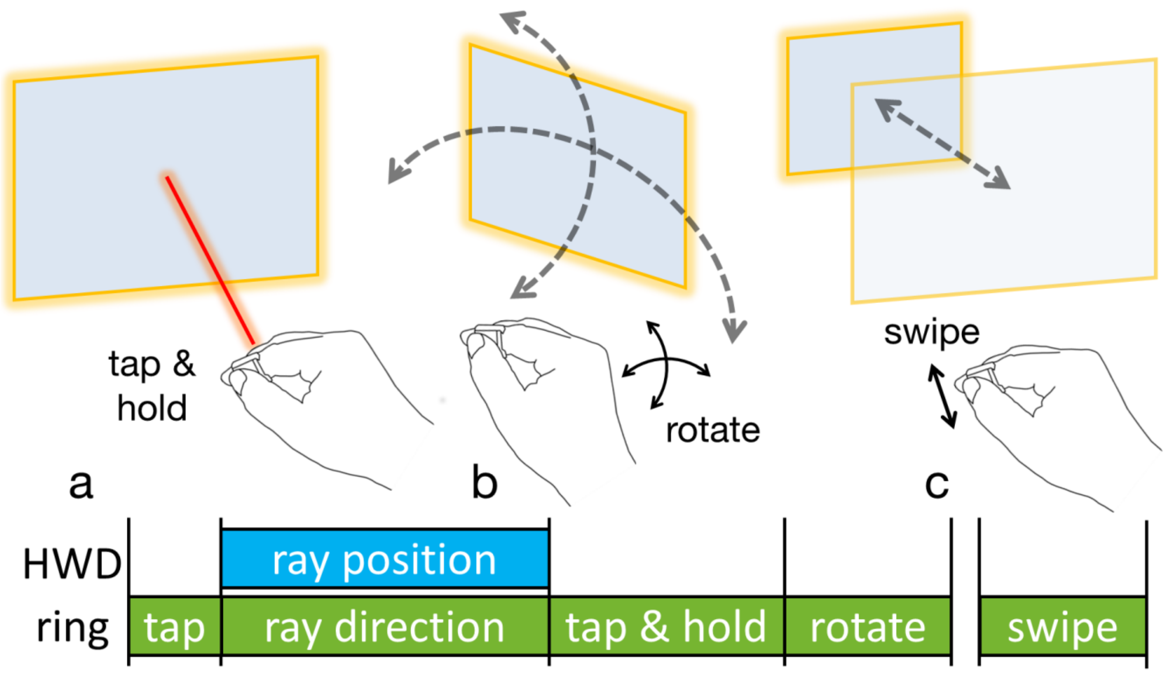

Figure 42. SAI Indirect Ring Input
Natural pointing gestures are useful when objects are out of reach. In this figure, a user repositions a window using the 'Ray-Grabbing' technique [30], which constrains the window movement. a) The user taps and holds to select a window. b) Wrist rotation controls the window's movement on the imaginary surface of a body-centric sphere. c) The window's depth can be manipulated using up and down swipes on the ring's touch pad.

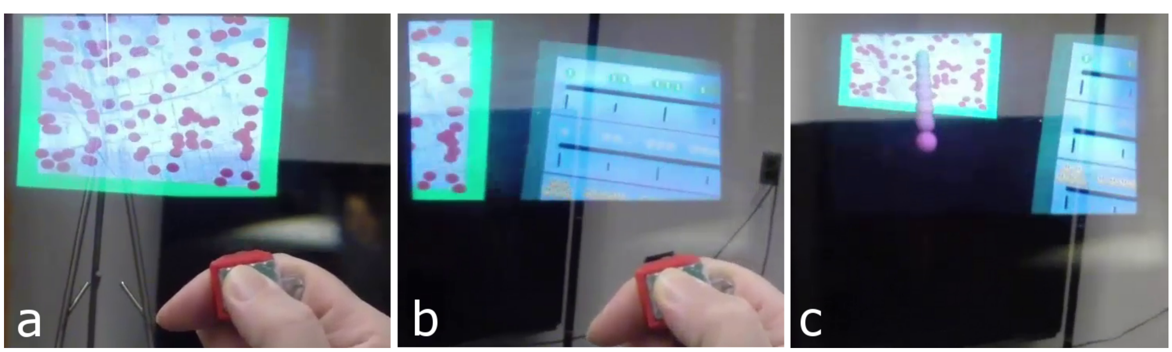

Figure 43. SAI Implementation: Indirect Ring Input
Raycasting is particularly useful to interact with objects that are out of reach. In this figure, a user grabs (a) a window (the virtual ray is hidden, but maintains the connection), and places it alongside another window (b). The window is then moved away in depth (c) using vertical swipe gestures.

6.6 Conclusion and Future Work

We have introduced a method for augmenting natural direct input with indirect input techniques by supplementing a wearable hand tracker with input from a networked ring device. Our implementation provides a proof-of-concept demonstration of several novel interaction techniques that combine two previously disjoint input technologies.

In future, we would like to explore a greater variety of interaction techniques, and explore new opportunities to exploit the combined benefits of hand and ring input. These developments can draw further from the VR interaction literature and integrate existing techniques with natural input methods.

One interesting possibility we have yet to fully explore is the idea of using the boundary of the depth camera frustum as an implicit trigger for input mode-switching. Direct manipulation and pointing techniques are used primarily in the user's line of sight, and ring input can be used while the arm is at rest, out of the camera's view; thus, the depth camera frustum is a natural boundary for triggering implicit actions to reduce switching costs.

There are other possible input devices that could support SAI interaction besides the ring we use in this work. Likewise, there are a variety of natural input methods that may be practical for use with wearables and HWDs. For instance, it would be interesting to explore the many possible interactions that could be created using only gaze input from an eye-tracking device coupled with a wearable device containing only a single button.

Of course, our developments will benefit from the input of test users. After sufficient refinement, these implementations would benefit from evaluation using realistic, data-driven, analytic tasks. Metrics such as time efficiency, precision, and fatigue will determine how the combination of input technologies we propose in this work will fare against the best existing input technologies for HWDs. Ultimately, we hope these techniques will prove helpful in providing productive user interfaces for wearables and HWDs as the next generation of mobile technologies.

7 Conclusions

This dissertation is the first body of research, to our knowledge, to provide a detailed exploration of interactive, multi-window interfaces to support in-situ, analytic tasks for wearable computing. We chose to follow a broad approach, addressing a wide range of requirements for such interfaces, rather than focusing in-depth on a single branch of the larger subject. This strategy supported a holistic introduction of the central concept of SAIs, and allowed us to investigate its various nuances sufficiently to meet our research objective as stated in the introduction of this work:

Investigate the primary design aspects of spatial interfaces that fulfill the major design requirements to support in-situ, analytic tasks on future wearable computing devices.

While we have provided an initial discussion of SAIs that is broad and thorough enough to warrant the naming of this concept as the title of this dissertation, we have in actuality only scratched its surface. A great deal of further work is required to flesh out the ideas we propose and to address the many subtleties and nuances of such interfaces that will make them useful and productive for use with real-world tasks.

In this final chapter, we provide a summary of our findings, and discuss several assumptions and limitations of this work. In exposing these limitations, we aim to shed light on the many future opportunities for building on this work. We then state our final conclusions with hope of inspiring future research.

7.1 Summary

This dissertation proposes the concept of SAIs and provides an introductory exploration of how spatial interfaces might be used to support in-situ, analytic taskwork on future wearable devices. To develop this concept, we begin by proposing a concise set of requirements which we draw from existing research on visualization and visual analytics and filter through the lens of work on spatial interfaces and wearable computing. Our introduction also contains a set of scenarios that help to portray our overall vision of how spatial interfaces can be used to support in-situ tasks, and to provide an initial glimpse of why requirements such as mobility and integration are important to consider in designs for users conducting everyday, analytic tasks.

Along with the introduction of SAIs, we provide a list of requirements drawn from related literature. The chapters of this thesis provide in-depth design explorations of various components of SAIs that touch on all the requirements in this list. We explore layouts of **multiple views**, in body-centric configurations to support **mobility**, and world-fixed configurations that allow **integration** in the surroundings. These layouts are designed to support ease of **interpretation**, allowing effective consumption of information. Finally, we explore **interactivity** in these environments to show how SAIs can support human-in-the-loop analytic taskwork. Moreover, we highlight some of the close interdependencies between these requirements throughout these discussions.

In chapter 2, we conduct a survey of prior work on spatial user interfaces, covering a wide variety of design concepts, technologies and implementations. From a close inspection of several dozen designs, alongside our consultation of many other seminal works and surveys on spatial interaction and 3D user interfaces, we parse a succinct design space consisting of seven design dimensions. We learn that our proposed design dimensions are not fully orthogonal, but contain several interdependencies that define broader dimensional categories and help to highlight several design decisions and strategies behind the concepts. We then show how projecting the dimensions back onto the original designs can help to categorize them, and in some cases reveal similarities that are not obvious at first glance.

Chapter 3 provides a brief overview of several classes of technology commonly used to implement spatial interfaces. We explain our decision to focus on interfaces for near-future HWDs in this dissertation by outlining in detail how this class of devices is particularly suitable for meeting our list of requirements for SAIs. In addition, we introduce the Ethereal Planes metaphor, on which the designs we present in the remaining chapters are based. We explain the reasoning behind our decision to pursue this paradigm of multiple 2D information spaces situated in 3D space, and contrast it with alternate possibilities for designing AR interfaces.

Chapter 4 reports the results of an in-depth series of user studies that investigate the human factors associated with many design parameters of a body-centric, spatial interface. To abstract the challenges and limitations of designing such an interface on current hardware, we emulate the limited viewing field of a HWD in a high-fidelity, low-latency CAVE environment with an external tracking system. From our results we propose the design for a specific body-centric layout, optimized for direct input by centering the layout around the user's dominant shoulder joint. Our final study compares this layout against two view-fixed designs that employ abstract navigation techniques, such as panning and icon selection, borrowed from modern smartphone interfaces. Our spatial interface allows study participants to complete a multi-display analytic task dramatically faster than both baseline interfaces, and was preferred by a majority. These results contribute to a growing body of evidence that spatial interfaces leveraging body motion can have distinct benefits over traditional abstract interfaces in the completion of analytic taskwork. We show that the benefit of natural head motion persists even with the limited viewing field of contemporary HWDs; this encourages our further exploration of SAIs in the subsequent chapters.

In chapter 5, we explore world-fixed layouts of spatial interfaces and their transitions from body-centric layouts. We confront questions about how to integrate content into various environments without impeding users' view of important real-world objects. Our implementation demonstrates how opposing goals such as spatial constancy and visual saliency can be balanced within a constraint-based layout algorithm. We conduct a user study to verify the intended results of our layout approach and empirically show that it can simultaneously reduce interference with salient regions and increase the predictability of items' layout locations. Qualitative findings produce user preferences such as alignment of edges and reduction of 'lonely' windows that can be used to inform the design of additional layout constraints.

Chapter 6 addresses the need for interaction in SAIs for performing analytic tasks. We first introduce the concept of dual-tier interaction in the Ethereal Planes metaphor. This concept encapsulates the need for supporting interaction at both layout and content layers, and for seamless switching between the two. As one potential solution for dual-tier interaction, we present an implementation that combines vision-based hand tracking with a minimalist ring-format input device. Our proof-of-concept demonstration shows how a variety of interaction techniques can be provided with this implementation in a wearable form factor. Our solution meets several criteria, such as mobility, precision selection and low fatigue, necessary for in-situ analytic taskwork.

The breadth of these explorations together demonstrate the potential of SAIs for meeting users' everyday needs. Our user studies, design explorations, and prototype implementations reveal it is possible to produce interfaces that meet the primary design requirements for SAIs, while satisfying the accompanying constraints of known human factors and of next-generation technologies.

Arc of Prototype Development

Throughout this work, we developed a variety of prototypes that incorporated a wide range of technologies. These developments began with a high-level concept, iteratively refined toward a working interface prototype HWD platform including additional prototype hardware. We see this arc of prototype development as a contribution in itself that may potentially benefit future research on new interface designs for technologies at the cusp of commercialization. Our methods are heavily inspired by visionary researchers such as Mark Weiser and Ivan Sutherland. Their seminal work led to new research techniques and workflows that are now common practice in the HCI research community. For example, future-casting, the projection of technological trends, allows the creation of vignettes that envision the problems future users will face and how yet-unrealized technologies might solve them. This method leads to emulation of future environments by substituting specialized existing technologies for those that will become ubiquitous.

This dissertation serves as an example of how such methods can be applied holistically to a broad problem and modified over the course of multiple interrelated projects conducted over a span of several years. The arc of our prototype development (**Figure 44**) follows several stages. Early in our work, we leveraged a CAVE system to provide reliable data about human factors needed to inform whether to invest further into our proposed design solution. In later work, we began moving to implementations on currently available hardware. For example, in Chapter 5 we began by using the HWD as only an external display tethered to a desktop computer to provide the necessary computational power.

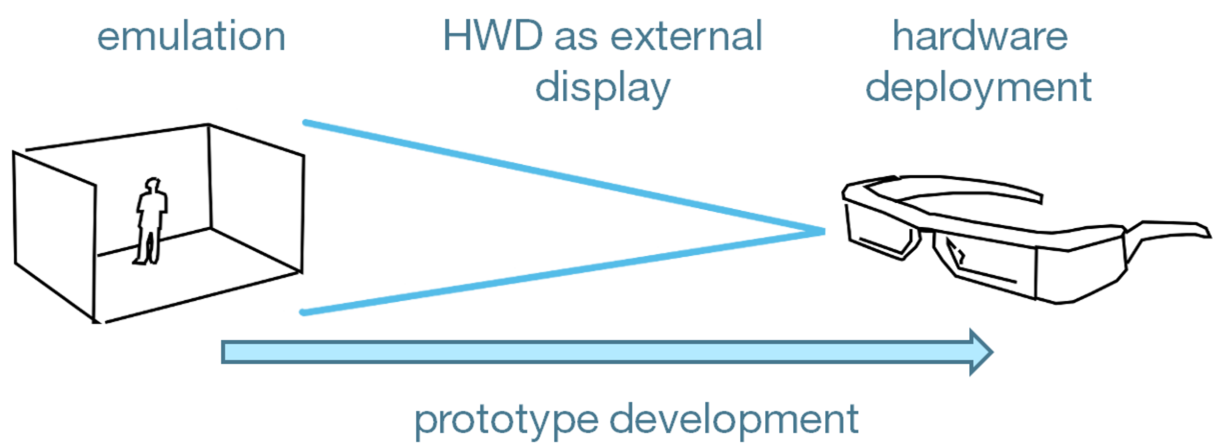

Figure 44. Arc of Prototype Development
We followed an arc of prototype development from our first emulated user studies toward a final hardware-based prototype design. The early stages of this research (Chapter 4) used a CAVE display to provide high fidelity sensing for investigating initial questions about human factors. Our later prototypes moved toward deployment on HWD technology, first as a tethered display (Chapter 5) and then as an independent platform (Chapter 6).

However, by the final stage of this work, our prototype was implemented on fully-wearable hardware components with the exception of the depth camera's hand-tracking server, which ran on a tethered computer. All other components were implemented on wearable components connected by wireless communication networks and compiled for Arduino board and the Android on the Moverio HWD unit.

The CAVE emulation and qualitative methods used in the early stages of this research provided high internal validity to answer the research questions we posed. To complement these methods, the shift toward standalone implementations in the later stages provides an increase in ecological validity. These prototype developments are accompanied by increasingly detailed discussion of the interface design and particulars about how it will work in real-world settings, but come with a tradeoff in the control and accuracy required to conduct rigorous user studies. This trajectory is holistically bound by our defined list of primary design requirements and motivated through each step by the future scenarios we presented in the introduction. These scenarios and our original concept were launched from the foundation of our design space that we created from the output of our initial, thorough literature review.

7.2 Assumptions and Limitations

Approaching such a broad topic as SAIs predictably requires a number of assumptions and comes with certain limitations. In this section, we discuss these assumptions and limitations and explore the primary ways they may be addressed in future work. We begin with the assumptions and limitations of each individual chapter, then discuss some broader challenges that must be addressed for our vision of SAI to grow toward practical implementations on future wearables.

Chapter 2 touched on the limitations of our design space for spatial interfaces. The literature survey is not exhaustive, nor is our set of proposed dimensions complete. Further exploration of this area may lead to the proposal of additional dimensions. In the chapter's conclusion, we single out multi-modal interfaces, co-located collaboration and extensions into 3D space as potential areas of interest. These are all pertinent to current developments in spatial interfaces and interactive AR in general. Multi-modal interfaces, for instance, are highly relevant to our discussion on interaction in chapter 6. This area has recently been explored for supporting natural interaction methods [118,162] and may provide alternative means for effective dual-tier interaction in SAIs. Further expansions of SAIs using 3D interfaces is inevitable. Interesting questions are ripe for exploration of when to use 2D vs 3D visualizations and how to combine and balance these contrasting modes in practical-use cases. We single out collaboration (including social, multi-user, etc.) as perhaps the most overlooked aspect of spatial interaction and we dedicate a portion of our discussion to this topic in the following section.

In chapter 3 we discuss the benefits of HWDs for SAIs and narrow the scope of our subsequent design explorations to this single technology. This step is necessary to make the dissertation tractable, but also highlights a promising technology for future consideration. In reality, however, it is likely that spatial interfaces will leverage a combination of technologies, including HWDs, projection, handheld devices and other potential technologies. As they become ubiquitous, each will have features that can be exploited for maximum benefit. For example, we may soon find projection-based displays in public spaces such as workplaces and shops, in which case we can develop applications that are specifically designed for particular locations. HWDs, meanwhile, can be used in environments where projectors are too costly or impractical. Handheld devices can supply tangible surfaces for interaction where these are most beneficial. These technologies can also be used in conjunction to overcome trade-offs. Projectors, being public by nature, can be used to display shared information, while HWDs supplement the same space with personal details.

Our series of cascading studies in chapter 4 makes some assumptions about parameters of the emulated hardware, such as FoV size and depth of the virtual display plane. The results of similar studies performed on real-world hardware would likely differ among different devices. For instance, the ideal distance for direct input may be influenced by effects of the virtual display depth as illustrated in **Figure 12**, and task productivity is likely to increase with FoV, since larger application windows will be capable of holding more information without breaching the viewing region boundaries. Nonetheless, we feel these studies provide a useful example for future researchers approaching similar questions on a variety of hardware platforms and device variations. We believe our primary result from this chapter, that spatial interfaces leveraging head motion can prove effective despite FoV constraints, will generalize to other setups and tasks. However, this generalization remains to be shown through further study of spatial interfaces.

We make several assumptions in chapter 5 in our concept and implementation about how a world-fixed spatial window manager should operate. Foremost, we assume a primary viewing pose exists for the user within each environment. This assumption is true for many environments, however, alternate solutions remains to be found for certain situations, particularly large rooms capable of holding many people and dynamic areas where people frequently move about. Second, we assume the existence of a fixed set of a small number of applications, for which the same layout is desired in all situations. In reality, people are likely to prefer several default configurations for different situations. These are likely to be dependent on context and location. For example, users preparing food in a kitchen may expect items such as recipes and cooking videos unlikely to be used in any other context. Third, we consider only static environments, where windows remain fixed once they are laid out. Real-world users will often find themselves in dynamic, bustling places, and solutions must be found for dealing with content placement in such environments. Finally, our work does not address situations where users are not stationary but would nonetheless prefer a world-fixed layout over a body-centric one. We believe that, in general, it is preferable for virtual content to have a fixed spatial relationship with real-world objects when possible. At present, the consequences of having objects follow users by sliding along walls or by jumping from object to object are unclear. Despite these uncertainties, these assumptions were necessary for our research to focus on the balancing of constraints within the layout algorithm.

Our look at interaction in chapter 6 is dependent on the particular style of interface we chose, being a spatial arrangement of 2D planes as proposed in previous chapters. Interaction styles and techniques may change substantially between different interface designs; however, our literature review in chapter 2 and results in chapter 4 provide evidence that the Ethereal Planes metaphor is highly suitable for analytic tasks. The primary limitation of our work in chapter 6 is the absence of a user evaluation. As discussed in the introduction and the previous section, introducing hardware prototypes comes with trade-offs of internal, experimental validity versus external, ecological validity. For example, the studies in chapter 4 were aimed at exploring the underlying human factors in task switching, and the study in chapter 5 was aimed at evaluating our layout manager. The evaluations of these studies were based on performance-based metrics, such as task time and user fatigue, hence, for these we developed prototypes with emulated and tethered displays, both of which relied on high-precision tracking systems. In contrast, chapter 6 is in part aimed at demonstrating a SAI on an actual wearable platform. This platform poses challenges for evaluation, such as drift, which is inherent in IMU-based head tracking, limited range of the depth camera, and low robustness of current hand-tracking methods. Therefore, we place our primary efforts on demonstrating how the proposed combination of technologies can support a wide variety of interaction techniques, and in showing how these can be interleaved to support seamless dual-tier interaction with SAI windows and contents.

With the breadth of this work, we set the stage for a holistic evaluation of SAIs. This will require the development of more advanced prototypes that can be used in longitudinal studies, to determine whether such systems can enhance user cognition and benefit actual in-situ tasks. As we conclude this dissertation, we hope our work will inspire future researchers and product developers to explore the many particulars of designing, implementing and testing various interaction techniques and features of SAIs.

7.3 Some Areas Deserving Future Work

In this section we highlight a few specific areas of particular importance to the development of SAIs. These have received some attention but are currently open areas of research in relation to the concepts presented in this dissertation. As the technology that will enable SAIs matures, realizing the benefits of such interfaces will depend on collaboration between experts in these and other areas of research.

Social and collaborative experiences

In addition to the set of primary requirements outlined in the introductory chapter, another important consideration for SAIs is the need for these wearable interfaces to fit into the users' social spheres. A fundamental and defining difference of spatial user interfaces from traditional interfaces is that interaction takes place in 3D space of the physical world rather than within the confines of 2D screen space. As a consequence, users will interact in a space that is shared by real-world physical objects and other people. As computing activities increasingly move away from the desktop and beyond the confines of glass screens, interface designers must keep in mind a twofold question: First, how is a person's interaction affected by others nearby; and second, how does their interaction affect others? Friends, family and passers-by pose distractions or various social constraints on the user's behaviour. The situation becomes even more complex when we consider that each actor in a social scene may simultaneously play both an interface user and an observer of others. If information is exchanged between users, then the interplay can become very complex according to various social nuances and a multitude of potential environments, combinations of participants and social contexts [57].

However, like much other interaction research, this dissertation contains a substantial limitation that our design spaces, user studies and prototype implementations all concern single persons in isolation. This assumption of single-user use cases is made for practical reasons. First, lab studies commonly abstract tasks and interactions into conceptual components. Also, studies on multi-user or collaborative interfaces often use different metrics and evaluation methods than those used in single-user studies. For example, the studies on collaboration may attempt to measure the success of information sharing [148] or the quality of shared experiences [112], whereas single user studies typically focus on time, error rate, and other such tractable metrics. In addition to these differences, there are practical challenges in coordinating studies with multiple users, such as extra cost and scheduling constraints. It may also be very difficult to observe natural social interactions between users in a controlled environment.

Despite these challenges there is also growing interest in research on social interaction, such as the StudentLife project of Wang et al. [222] that uses ubiquitous smartphone sensors to collect information to help improve the health and mental well-being of college students. Other work is exploring how interfaces can be designed to support group interaction, for example by recognizing and adapting to group configurations known as 'F-formations' [133,134]. Physiological sensors in wearable devices are also being used to determine people's emotions and to incorporate this information into shared experiences [13,20]. For the field of spatial interaction to mature, it is important

– and inevitable – that the lines blur between spatial and social interaction, and that experts in these fields bridge their work into holistic systems.

Tangible Interaction

One drawback of virtual interfaces lies in their absence of physicality. As we noted in chapter 3, the benefit of HWDs to superimpose virtual content in empty space comes with a trade-off of intangibility. Projection-based and handheld technologies do not face this problem; however, they are necessarily restricted to placing content on existing surfaces. To build an ultimate interface, as envisioned by Sutherland [196], would require eliminating this trade-off altogether, providing virtual objects with physical substance. Unfortunately, this idea, at least for now, lies in realm of science fiction (i.e. the famed Holodeck from Star Trek: The Next Generation). Nonetheless, there are efforts underway to approach this challenging problem.

As we noted in our design-space exploration in chapter 2, one strategy for mitigating the drawbacks of intangible interfaces is the provision of palette-like objects for use in virtual settings [82,183,199]. Another influential technique, introduced by Billinghurst et al. [24], uses fiducial makers to act as both tracking support and proxy objects for virtual counterparts. This technique is now widely used in HAR interfaces, including commercial promotions using mobile applications such as Blippar [25]. More recently, researchers have explored more far-reaching possibilities for tangible interaction in virtual environments. The whimsical 'Haptic Turk' metaphor introduced by Cheng et al. [43] uses human actors to provide physical props that coincide with virtual objects felt by users in a VR environment. A similar idea uses a robotic arm that dynamically moves to present a haptic surface at the user's predicted touch location [8]. One promising area of work is 'ultrahaptics', introduced by Carter et al. [37,227], which uses focused ultrasonic sound waves to provide a sense of touch in thin air. In contrast to these elaborate mechanisms, a simpler strategy for mobile interfaces is to appropriate objects found in the environment for the situation of haptic interfaces [90].

The developments outlined here portray the variety of imaginative methods that researchers have developed to overcome the seemingly intractable problem of touching virtual interfaces. The mixture of tangible and spatial interfaces is fairly new, but this discussion highlights the need for more work in this area to overcome one of the greatest deficiencies of spatial user interfaces.

Information Visualization and Visual Analytics

The one primary requirement for SAIs of the five we introduced in this work – and the one to receive the least attention in this dissertation – is the requirement for interpretation. We discussed how SAIs can support the division of information into digestible parts, and shifting of attention between these using head motion. We also discussed some specific design considerations, such as keeping specific items spatially constant, or placing all items related to a single task fully within view when possible. However, the question of how to best portray information that populates the window contents deserves a great deal of consideration.

In defining SAIs, we have already incorporated some considerations drawn from the fields of information visualization and visual analytics. However, there is much room to grow in the relationship between these areas of research and spatial user interfaces. So far, there is little work that bridges this gap, aside from that covered in this document. Here again, we emphasize the need for researchers from multiple disciplines to collaborate in the development of SAIs.

Here we leave a few final thoughts about open design challenges for visualizing information in SAI windows. First, whereas visual analytics is primarily aimed at expert users conducting complex tasks in a specific knowledge domain, our vision for SAIs targets typical users of everyday applications. Therefore, it is important to design visualizations that are easy to use and immediately understandable with little or no explicit training. Another important consideration for designing visualizations is to make them appropriate for the specific platform used. In the case of HWDs, this means limiting the complexity of information to what can be practically supported by the display. For example, current HWDs have relatively limited resolution and FoV, so the information presented in a single window should be depicted with large enough images and labels, yet be small enough to fit within the window. These restrictions place a relatively low cap on the information bandwidth for certain devices. Visual content should also consider the particular use case of the device. For instance, visualizations designed for mobile use cases in the Personal Cockpit should be relatively easy to interpret to avoid undue distraction. Likewise, ambient information displays [167] embedded in a particular environment should not be overly complex. Integrating information into existing environments will also require some expertise, as it should blend in well to prevent unnecessary distraction.

7.4 A Final Word

In the previous pages, we outline a new class of interface that applies the benefits of spatial interaction for analytic taskwork for in-situ use cases. We propose that wearable interfaces can be made to be much more powerful, useful and productive than in their current form of supporting micro-interactions. However, it is not the goal of this work to supersede current forms of computing, such as smartphones and desktop computing. Rather, we aim to highlight the advantages of the SAI paradigm in certain cases. For example, current mobile devices necessitate focusing at a close distance for extended periods, whereas SAIs can place content at a more comfortable distance, when circumstances allow. Smartphones and tablet computers nonetheless retain the advantage of haptic surfaces, and may remain in use for particular applications, much as pens and clipboards have not been supplanted by computer interfaces in all cases. Likewise, desktop computers will remain pillars of productivity for some time to come. However, as technology and interface designs progress, SAIs will increasingly mimic the benefits of desktop computers while also allowing the advantages of mobility and in-situ use. We hope that our primary contribution in the introduction of SAIs and our many additional contributions – of outlined requirements, design spaces and explorations, quantitative and qualitative studies, in-depth analysis and prototype implementations – will help to usher forth a new era of useful, productive, and engaging in-situ, wearable computing.

References

1. Anand Agarawala and Ravin Balakrishnan. 2006. Keepin' it real: Pushing the desktop metaphor with physics, piles and the pen. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '06), 1283–1292. http://doi.org/10.1145/1124772.1124965
2. Maneesh Agrawala, Andrew C. Beers, Ian McDowall, Bernd Fröhlich, Mark Bolas, and Pat Hanrahan. 1997. The two-user responsive workbench: Support for collaboration through individual views of a shared space. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques* (SIGGRAPH '97), 327–332. http://doi.org/10.1145/258734.258875
3. Patricia L. Alfano and George F. Michel. 1990. Restricting the field of view: Perceptual and performance effects. *Perceptual and motor skills* 70, 1: 35–45.
4. Fraser Anderson, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2015. Supporting subtlety with deceptive devices and illusory interactions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 1489–1498.
5. Android Wear | Android Developers. Retrieved April 21, 2016 from http://developer.android.com/wear/index.html
6. Carlos Andujar and Ferran Argelaguet. 2006. Friction Surfaces: Scaled ray-casting manipulation for interacting with 2D GUIs. In *Proceedings of the 12th Eurographics Conference on Virtual Environments* (EGVE'06), 101–108. http://doi.org/10.2312/EGVE/EGVE06/101-108
7. Dennis R. Ankrum. 1999. Visual ergonomics in the office -- Guidelines. *Occupational Health and Safety* 68, 7: 64–74.
8. Bruno Araujo, Ricardo Jota, Varun Perumal, Jia Xian Yao, Karan Singh, and Daniel Wigdor. 2016. Snake Charmer: Physically enabling virtual objects. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction* (TEI '16), 218–226. http://doi.org/10.1145/2839462.2839484
9. Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics* 37, 3: 121–136. http://doi.org/10.1016/j.cag.2012.12.003
10. Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nenya: Subtle and eyes-free mobile input with a magnetically-tracked finger ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11), 2043–2046. http://doi.org/10.1145/1978942.1979238
11. Daniel Lee Ashbrook. 2010. Enabling mobile microinteractions. Ph.D. Dissertation. Georgia Institute of Technology, Atlanta, GA, USA. Advisor(s) Thad E. Starner.
12. Daniel Avrahami, Jacob O. Wobbrock, and Shahram Izadi. 2011. Portico: Tangible interaction on and around a tablet. In *Proceedings of the 24th Annual ACM*

Symposium on User Interface Software and Technology (UIST '11), 347–356. http://doi.org/10.1145/2047196.2047241

13. Sudhanshu S.D.P. Ayyagari, Kunal Gupta, Matt Tait, and Mark Billinghurst. 2015. CoSense: Creating shared emotional experiences. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems* (CHI EA '15), 2007–2012. http://doi.org/10.1145/2702613.2732839
14. Sungmin Bae, Yong Cheol Peter Cho, Sungho Park, Kevin M. Irick, Yongseok Jin, and Vijaykrishnan Narayanan. 2011. An FPGA implementation of information theoretic visual-saliency system and its optimization. In *IEEE 19th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM '11)*, 41–48. http://doi.org/10.1109/FCCM.2011.41
15. Michelle Q. Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. 2000. Guidelines for using multiple views in information visualization. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (AVI '00), 110–119. http://doi.org/10.1145/345513.345271
16. Robert Ball and Chris North. 2008. The effects of peripheral vision and physical navigation on large scale visualization. In *Proceedings of Graphics Interface 2008* (GI '08), 9–16.
17. Lyn Bartram, Johnny Rodgers, and Rob Woodbury. 2011. Smart homes or smart occupants? Supporting aware living in the home. In *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part II* (INTERACT '11), 52–64.
18. Blaine A. Bell and Steven K. Feiner. 2000. Dynamic space management for user interfaces. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology* (UIST '00), 239–248. http://doi.org/10.1145/354401.354790
19. Blaine Bell, Steven Feiner, and Tobias Höllerer. 2001. View management for virtual and augmented reality. In *Proceedings of the 14th annual ACM symposium on User interface software and technology* (UIST '00), 101–110.
20. Mark Billinghurst. 2014. Using augmented reality to create empathic experiences. In *Proceedings of the 19th International Conference on Intelligent User Interfaces* (IUI '14), 5–6. http://doi.org/10.1145/2557500.2568057
21. Mark Billinghurst, J Bowskill, N. Dyer, and J. Morphett. 1998. An evaluation of wearable information spaces. In *Virtual Reality Annual International Symposium, 1998. Proceedings. IEEE*, 20–27. http://doi.org/10.1109/VRAIS.1998.658418
22. Mark Billinghurst, Jerry Bowskill, Mark Jessop, and Jason Morphett. 1998. A wearable spatial conferencing space. In *Proceedings of the 2nd IEEE International Symposium on Wearable Computers* (ISWC '98), 76-83.
23. Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. 2001. The MagicBook: a transitional AR interface. *Computers & Graphics* 25, 5: 745–753. http://doi.org/10.1016/S0097-8493(01)00117-0
24. Mark Billinghurst, Hirokazu Kato, and Ivan Poupyrev. 2008. Tangible augmented reality. *ACM SIGGRAPH ASIA,* 7.
25. Blippar. Retrieved April 23, 2016 from https://blippar.com/en/

26. Matt R. Bohm, Jayson P. Vucovich, and Robert B. Stone. 2008. Using a design repository to drive concept generation. *Journal of Computing and Information Science in Engineering* 8, 1: 14502. http://doi.org/10.1115/1.2830844
27. Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. 2010. Touch Projector: Mobile interaction through video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 2287–2296. http://doi.org/10.1145/1753326.1753671
28. Jean Yves Bouguet. Camera calibration toolbox for matlab. Retrieved April 19, 2016 from http://www.vision.caltech.edu/bouguetj/calib_doc/
29. Doug A. Bowman (ed.). 2005. *3D User Interfaces: Theory and Practice*. Addison-Wesley, Boston.
30. Doug A. Bowman and Larry F. Hodges. 1997. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics* (I3D '97), 35–ff. http://doi.org/10.1145/253284.253301
31. Doug A. Bowman and Larry F. Hodges. 1999. Formalizing the design, evaluation, and application of interaction techniques for immersive virtual environments. *Journal of Visual Languages & Computing* 10, 1: 37–53. http://doi.org/10.1006/jvlc.1998.0111
32. Doug A. Bowman, Ryan P. McMahan, and Eric D. Ragan. 2012. Questioning naturalism in 3D user interfaces. *Commun. ACM* 55, 9: 78–88. http://doi.org/10.1145/2330667.2330687
33. Neil Bruce and John Tsotsos. 2006. Saliency based on information maximization. In *Advances in Neural Information Processing Systems 18*, Y. Weiss, B. Schölkopf and J. C. Platt (eds.). MIT Press, 155–162.
34. Alex Butler, Shahram Izadi, and Steve Hodges. 2008. SideSight: Multi-"touch" interaction around small devices. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (UIST '08), 201–204. http://doi.org/10.1145/1449715.1449746
35. Bill Buxton and George W. Fitzmaurice. 1998. HMDs, Caves & Chameleon: A human-centric analysis of interaction in virtual space. *SIGGRAPH Comput. Graph.* 32, 4: 69–74. http://doi.org/10.1145/307710.307732
36. Xiang Cao and Ravin Balakrishnan. 2006. Interacting with dynamically defined information spaces using a handheld projector and a pen. In *Proceedings of the 19th Annual ACM Symposium on User interface software and technology* (UIST '06), 225–234.
37. Tom Carter, Sue Ann Seah, Benjamin Long, Bruce Drinkwater, and Sriram Subramanian. 2013. UltraHaptics: Multi-point mid-air haptic feedback for touch surfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (UIST '13), 505–514. http://doi.org/10.1145/2501988.2502018
38. Jessica Cauchard, Markus Löchtefeld, Mike Fraser, Antonio Krüger, and Sriram Subramanian. 2012. m+pSpaces: Virtual workspaces in the spatially-aware mobile environment. In *Proceedings of the 14th International Conference on Human-*

computer Interaction with Mobile Devices and Services (MobileHCI '12), 171–180. http://doi.org/10.1145/2371574.2371601
39. Moran Cerf, E. Paxon Frady, and Christof Koch. 2009. Faces and text attract gaze independent of the task: Experimental data and computer model. *Journal of Vision* 9, 12: 10–10. http://doi.org/10.1167/9.12.10
40. Li-Wei Chan, Hui-Shan Kao, Mike Y. Chen, Ming-Sui Lee, Jane Hsu, and Yi-Ping Hung. 2010. Touching the void: Direct-touch interaction for intangible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 2625–2634. http://doi.org/10.1145/1753326.1753725
41. Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, et al. 2013. FingerPad: Private and subtle interaction using fingertips. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (UIST '13), 255–260. http://doi.org/10.1145/2501988.2502016
42. Xiang "Anthony" Chen, Nicolai Marquardt, Anthony Tang, Sebastian Boring, and Saul Greenberg. 2012. Extending a mobile device's interaction space through body-centric interaction. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services* (MobileHCI '12), 151–160.
43. Lung-Pan Cheng, Patrick Lühne, Pedro Lopes, Christoph Sterz, and Patrick Baudisch. 2014. Haptic Turk: A Motion Platform Based on People. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (CHI '14), 3463–3472. http://doi.org/10.1145/2556288.2557101
44. CNET. How to use the Microsoft Surface touch screen and keyboard. Retrieved April 20, 2016 from http://www.cnet.com/how-to/how-to-use-the-microsoft-surface-touch-screen-and-keyboard/
45. Andy Cockburn, Philip Quinn, Carl Gutwin, Gonzalo Ramos, and Julian Looser. 2011. Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *International Journal of Human-Computer Studies* 69, 6: 401–414. http://doi.org/10.1016/j.ijhcs.2011.02.005
46. Christopher M. Collins and Sheelagh Carpendale. 2007. VisLink: Revealing relationships amongst visualizations. IEEE *Transactions on Visualization and Computer Graphics,* 13, 6: 1192–1199.
47. Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. 1993. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques* (SIGGRAPH '93), 135–142. http://doi.org/10.1145/166117.166134
48. Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. DeFanti, Robert V. Kenyon, and John C. Hart. 1992. The CAVE: Audio visual experience automatic virtual environment. *Commun. ACM* 35, 6: 64–72. http://doi.org/10.1145/129888.129892
49. Mary Czerwinski, Desney S. Tan, and George G. Robertson. 2002. Women take a wider view. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '02), 195–202. http://doi.org/10.1145/503376.503412
50. Stephen DiVerdi, Daniel Nurmi, and Tobias Höllerer. 2003. ARWin-A desktop augmented reality window manager. In *Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality* (ISMAR '03), 298.

51. Tanja Döring, Axel Sylvester, and Albrecht Schmidt. 2013. A design space for ephemeral user interfaces. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction*. ACM, 75–82.
52. Lorin Elias and Deborah Saucier. 2005. *Neuropsychology: Clinical and Experimental Foundations*. Pearson, Boston.
53. Niklas Elmqvist and Pourang Irani. 2013. Ubiquitous Analytics: Interacting with big data anywhere, anytime. *Computer 46*, 86–89.
54. Neven A. M. ElSayed, Bruce H. Thomas, Ross T. Smith, Kim Marriott, and Julia Piantadosi. 2015. Using augmented reality to support situated analytics. In *2015 IEEE Virtual Reality* (VR '15), 175–176. http://doi.org/10.1109/VR.2015.7223352
55. Barrett Ens, Ahmad Byagowi, Teng Han, Juan David Hincapié-Ramos, and Pourang Irani. Combining ring input with hand tracking for precise, natural interaction with spatial analytic interfaces. In *Proceedings of the 4th ACM Symposium on Spatial User Interaction* (SUI '16), 4 pages, to appear.
56. Barrett Ens, Rory Finnegan, and Pourang P. Irani. 2014. 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* (CHI '14), 3171–3180. http://doi.org/10.1145/2556288.2557058
57. Barrett Ens, Tovi Grossman, Fraser Anderson, Justin Matejka, and George Fitzmaurice. 2015. Candid Interaction: Revealing hidden mobile and wearable computing activities. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (UIST '15), 467–476. http://doi.org/10.1145/2807442.2807449
58. Barrett Ens, Juan David Hincapié-Ramos, and Pourang Irani. 2014. Ethereal Planes: A design framework for 2D information space in 3D mixed reality environments. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction* (SUI '14), 2–12. http://doi.org/10.1145/2659766.2659769
59. Barrett Ens and Pourang Irani. 2016. Spatial Analytic Interfaces: Spatial user interfaces for in-situ visual analytics. *IEEE Computer Graphics and Applications*, 8 pages, to appear. http://doi.org/10.1109/MCG.2016.38
60. Barrett Ens, Eyal Ofek, Neil Bruce, and Pourang Irani. Shared Façades: Surface-embedded layout management for ad-hoc collaboration using head-worn displays. In *Collaboration Meets Interactive Surfaces and Spaces (CMISS) – Theory and Practice.* Craig Anslow, Pedro Campos, Joaquim Jorge, (Eds). Springer-Verlag, 8 pages, to appear.
61. Barrett Ens, Eyal Ofek, Neil Bruce, and Pourang Irani. 2015. Spatial constancy of surface-embedded layouts across multiple environments. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction* (SUI '15) 65–68. http://doi.org/10.1145/2788940.2788954
62. Epson America, Inc. Epson Moverio BT-300 and BT-200 Smart Glasses. Retrieved April 17, 2016 from http://www.epson.com/cgi-bin/Store/jsp/Landing/moverio-augmented-reality-smart-glasses.do
63. Steven Feiner, Blair MacIntyre, Marcus Haupt, and Eliot Solomon. 1993. Windows on the World: 2D windows for 3D augmented reality. In *Proceedings of the 6th*

Annual ACM Symposium on User Interface Software and Technology (UIST '93), 145–155. http://doi.org/10.1145/168642.168657
64. Scott S. Fisher, Michael McGreevy, James Humphries, and Warren Robinett. 1987. Virtual environment display system. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics* (I3D '86), 77–87. http://doi.org/10.1145/319120.319127
65. George W. Fitzmaurice. 1993. Situated Information Spaces and Spatially Aware Palmtop Computers. *Commun. ACM* 36, 7: 39–49. http://doi.org/10.1145/159544.159566
66. George W. Fitzmaurice, Shumin Zhai, and Mark H. Chignell. 1993. Virtual Reality for Palmtop Computers. *ACM Trans. Inf. Syst.* 11, 3: 197–218. http://doi.org/10.1145/159161.159160
67. Clifton Forlines, Daniel Vogel, and Ravin Balakrishnan. 2006. HybridPointing: Fluid switching between absolute and relative pointing with a direct input device. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (UIST '06), 211–220. http://doi.org/10.1145/1166253.1166286
68. Scott Frees, G. Drew Kessler, and Edwin Kay. 2007. PRISM Interaction for Enhancing Control in Immersive Virtual Environments. *ACM Trans. Comput.-Hum. Interact.* 14, 1. http://doi.org/10.1145/1229855.1229857
69. Jon Froehlich, Tawanna Dillahunt, Predrag Klasnja, et al. 2009. UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), 1043–1052.
70. Joseph L. Gabbard, J. Edward Swan, and Deborah Hix. 2006. The effects of text drawing styles, background textures, and natural lighting on text legibility in outdoor augmented reality. *Presence* 15, 1: 16–32. http://doi.org/10.1162/pres.2006.15.1.16
71. Ran Gal, Lior Shapira, Eyal Ofek, and Pushmeet Kohli. 2014. FLARE: Fast layout for augmented reality applications. In *2014 IEEE International Symposium on Mixed and Augmented Reality* (ISMAR '14), 207–212. http://doi.org/10.1109/ISMAR.2014.6948429
72. Juergen Gausemeier, Juergen Fruend, Carsten Matysczok, Beat Bruederlin, and David Beier. 2003. Development of a real time image based object recognition method for mobile AR-devices. In *Proceedings of the 2Nd International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa* (AFRIGRAPH '03), 133–139. http://doi.org/10.1145/602330.602355
73. Google Glass. Retrieved April 17, 2016 from http://www.google.com/glass/start/
74. Raphael Grasset, Tobias Langlotz, Denis Kalkofen, Markus Tatzgern, and Dieter Schmalstieg. 2012. Image-driven view management for augmented reality browsers. In *Proceedings of the 2012 IEEE International Symposium on Mixed and Augmented Reality* (ISMAR '12), 177–186. http://doi.org/10.1109/ISMAR.2012.6402555
75. Saul Greenberg, Michael Boyle, and Jason Laberge. 1999. PDAs and shared public displays: Making personal information public, and public information personal. *Personal Technologies* 3, 1–2: 54–64. http://doi.org/10.1007/BF01305320
76. Amiad Gurman. 2015. Depth mapping with enhanced resolution. Retrieved August 6, 2016 from http://www.google.ca/patents/US9019267

77. Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. 2010. Imaginary interfaces: spatial interaction with empty hands and without visual feedback. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology* (UIST '10), 3–12.
78. Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary Phone: Learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (UIST '11), 283–292. http://doi.org/10.1145/2047196.2047233
79. Sinem Guven, Steven Feiner, and Ohan Oda. 2006. Mobile augmented reality interaction techniques for authoring situated media on-site. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality* (ISMAR '06), 235–236. http://doi.org/10.1109/ISMAR.2006.297821
80. Taejin Ha, Steven Feiner, and Woontack Woo. 2014. WeARHand: Head-worn, RGB-D camera-based, bare-hand user interface with visually enhanced depth perception. In *IEEE International Symposium on Mixed and Augmented Reality* (ISMAR '14), 219–228.
81. Vicki Ha, Jim Wallace, Ryder Ziola, and Kori Inkpen. 2006. My MDE: Configuring virtual workspaces in multi-display environments. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '06), 1481–1486. http://doi.org/10.1145/1125451.1125723
82. Gerwin de Haan, Eric J. Griffith, Michal Koutek, and Frits H. Post. 2006. Hybrid interfaces in VEs: intent and interaction. In *EGVE*, 109–118.
83. Gerwin de Haan, Michal Koutek, and Frits H. Post. 2002. Towards intuitive exploration tools for data visualization in VR. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology* (VRST '02), 105–112. http://doi.org/10.1145/585740.585758
84. Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11), 441–450.
85. Chris Harrison and Scott E. Hudson. 2009. Abracadabra: Wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology* (UIST '09), 121–124. http://doi.org/10.1145/1622176.1622199
86. Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: Appropriating the body as an input surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 453–462. http://doi.org/10.1145/1753326.1753394
87. Khalad Hasan, David Ahlström, and Pourang Irani. 2013. AD-Binning: Leveraging around device space for storing, browsing and retrieving mobile device content. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 899–908. http://doi.org/10.1145/2470654.2466115
88. W. Keith Hastings. 1970. Monte Carlo sampling methods using markov chains and their applications. *Biometrika* 57, 1: 97–109. http://doi.org/10.2307/2334940
89. Jeffrey Heer and Ben Shneiderman. 2012. Interactive dynamics for visual analysis. *Queue* 10, 2: 30.

90. Anuruddha Hettiarachchi and Daniel Wigdor. 2016. Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 1957-1967.
91. Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), 1063–1072. http://doi.org/10.1145/2556288.2557130
92. Juan David Hincapié-Ramos, Levko Ivanchuk, Srikanth K. Sridharan, and Pourang P. Irani. 2015. SmartColor: Real-time color and contrast correction for optical see-through head-mounted displays. *IEEE Transactions on Visualization and Computer Graphics* 21, 12: 1336–1348. http://doi.org/10.1109/TVCG.2015.2450745
93. Juan David Hincapié-Ramos, Stephen Voida, and Gloria Mark. 2011. A design space analysis of availability-sharing systems. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (UIST '11), 85–96.
94. Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. 1994. A survey of design issues in spatial input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology* (UIST '94), 213–222.
95. Ken Hinckley, Jeff Pierce, Mike Sinclair, and Eric Horvitz. 2000. Sensing techniques for mobile interaction. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology* (UIST '00), 91–100. http://doi.org/10.1145/354401.354417
96. Thuong N. Hoang and Bruce H. Thomas. 2010. Augmented Viewport: An action at a distance technique for outdoor AR using distant and zoom lens cameras. In *International Symposium on Wearable Computers* (ISWC), 1–4.
97. David M. Hoffman, Ahna R. Girshick, Kurt Akeley, and Martin S. Banks. 2008. Vergence-accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision* 8, 3: 33–33. http://doi.org/10.1167/8.3.33
98. Nicholas P. Holmes and Charles Spence. 2004. The body schema and the multisensory representation(s) of peripersonal space. *Cognitive processing* 5, 2: 94–105. http://doi.org/10.1007/s10339-004-0013-3
99. HTC. Vive. Retrieved April 22, 2016 from https://www.htcvive.com/ca/
100. Dandan Huang, Melanie Tory, Bon Adriel Aseniero, et al. 2015. Personal visualization and personal visual analytics. *IEEE Transactions on Visualization and Computer Graphics* 21, 3: 420–433. http://doi.org/10.1109/TVCG.2014.2359887
101. Shahram Izadi, David Kim, Otmar Hilliges, et al. 2011. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (UIST '11), 559–568. http://doi.org/10.1145/2047196.2047270
102. Brett R. Jones, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. 2013. IllumiRoom: peripheral projected illusions for interactive experiences. In *ACM SIGGRAPH 2013 Emerging Technologies* (SIGGRAPH '13), 7:1–7:1. http://doi.org/10.1145/2503368.2503375

103. Brett R. Jones, Rajinder Sodhi, Pulkit Budhiraja, Kevin Karsch, Brian Bailey, and David Forsyth. 2015. Projectibles: Optimizing surface color for projection. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (UIST '15), 137–146. http://doi.org/10.1145/2807442.2807486
104. Brett Jones, Rajinder Sodhi, David Forsyth, Brian Bailey, and Giuliano Maciocci. 2012. Around device interaction for multiscale navigation. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services* (MobileHCI '12), 83–92. http://doi.org/10.1145/2371574.2371589
105. Shaun K. Kane, Daniel Avrahami, Jacob O. Wobbrock, et al. 2009. Bonfire: A nomadic system for hybrid laptop-tabletop interaction. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (UIST '09), 129–138. http://doi.org/10.1145/1622176.1622202
106. Youn-ah Kang and John Stasko. 2008. Lightweight task/application performance using single versus multiple monitors: A comparative study. In *Proceedings of Graphics Interface 2008* (GI '08), 17–24. Retrieved April 18, 2016 from http://dl.acm.org/citation.cfm?id=1375714.1375718
107. Karin Kappel and Thomas Grechenig. 2009. "Show-me": Water consumption at a glance to promote water conservation in the shower. In *Proceedings of the 4th international conference on persuasive technology*. ACM, 26. http://doi.org/10.1145/1541948.1541984
108. Hirokazu Kato and Mark Billinghurst. 1999. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *2nd IEEE and ACM International Workshop on Augmented Reality, 1999. (IWAR '99) Proceedings*, 85–94. http://doi.org/10.1109/IWAR.1999.803809
109. Bonifaz Kaufmann and Martin Hitz. 2012. X-large virtual workspaces for projector phones through peephole interaction. In *Proceedings of the 20th ACM International Conference on Multimedia* (MM '12), 1279–1280. http://doi.org/10.1145/2393347.2396441
110. Wolf Kienzle and Ken Hinckley. 2014. LightRing: Always-available 2D input on any surface. *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14). 157–160. http://doi.org/10.1145/2642918.2647376
111. Wolf Kienzle and Kenneth P. Hinckley. 2016. Smart ring. Retrieved August 6, 2016 from http://www.google.ca/patents/US20160077587
112. Seungwon Kim, Sasa Junuzovic, and Kori Inkpen. 2014. The nomad and the couch potato: Enriching mobile shared experiences with contextual information. In *Proceedings of the 18th International Conference on Supporting Group Work* (GROUP '14), 167–177. http://doi.org/10.1145/2660398.2660409
113. Georg Klein and David Murray. 2009. Parallel Tracking and Mapping on a camera phone. In *8th IEEE International Symposium on Mixed and Augmented Reality, 2009.* (ISMAR '09), 83–86. http://doi.org/10.1109/ISMAR.2009.5336495
114. Ernst Kruijff, J. Edward Swan, and Steven Feiner. 2010. Perceptual issues in augmented reality revisited. In *2010 9th IEEE International Symposium on Mixed and Augmented Reality* (ISMAR '10), 3–12. http://doi.org/10.1109/ISMAR.2010.5643530

115. Tobias Langlotz, Thanh Nguyen, Dieter Schmalstieg, and Raphael Grasset. 2014. Next-generation augmented reality browsers: Rich, seamless, and adaptive. *Proceedings of the IEEE* 102, 2: 155–169. http://doi.org/10.1109/JPROC.2013.2294255
116. Laster. *Laster Technologies*. Retrieved April 26, 2016 from http://www.laster.fr/en/
117. Gun A. Lee, Ungyeon Yang, Yongwan Kim, et al. 2009. Freeze-set-go interaction method for handheld mobile augmented reality environments. In *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology* (VRST '09), 143–146. http://doi.org/10.1145/1643928.1643961
118. Minkyung Lee, Mark Billinghurst, Woonhyuk Baek, Richard Green, and Woontack Woo. 2013. A usability study of multimodal input in an augmented reality environment. *Virtual Reality* 17, 4: 293–305. http://doi.org/10.1007/s10055-013-0230-0
119. Wonwoo Lee, Youngmin Park, Vincent Lepetit, and Woontack Woo. 2011. Video-based in situ tagging on mobile phones. *IEEE Trans. Cir. and Sys. for Video Technol.* 21, 10: 1487–1496. http://doi.org/10.1109/TCSVT.2011.2162767
120. Alex Leykin and Mihran Tuceryan. 2004. 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 '04), 224–230. http://doi.org/10.1109/ISMAR.2004.22
121. Frank Chun Yat Li, David Dearman, and Khai N. Truong. 2009. Virtual Shelves: Interactions with orientation aware devices. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology* (UIST '09), 125–128. http://doi.org/10.1145/1622176.1622200
122. Jiandong Liang and Mark Green. 1994. JDCAD: A highly interactive 3D modeling system. *Computers & Graphics* 18, 4: 499–506. http://doi.org/10.1016/0097-8493(94)90062-0
123. James Jeng-Weei Lin, Henry B. L. Duh, Habib Abi-Rached, Donald E. Parker, and Thomas A. Furness Iii. 2002. Effects of field of view on presence, enjoyment, memory, and simulator sickness in a virtual environment. In *Proceedings of the IEEE Virtual Reality Conference 2002* (VR '02), 164–171.
124. Robert W. Lindeman, John L. Sibert, and James K. Hahn. 1999. Towards Usable VR: An empirical study of user interfaces for immersive virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '99), 64–71. http://doi.org/10.1145/302979.302995
125. Shai Litvak, Leonid Brailovsky, and Tomer Yanir. 2015. Learning-based estimation of hand and finger pose. Retrieved August 6, 2016 from http://www.google.ca/patents/US9002099
126. Can Liu, Olivier Chapuis, Michel Beaudouin-Lafon, Eric Lecolinet, and Wendy E. Mackay. 2014. Effects of display size and navigation type on a classification task. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems* (CHI '14), 4147–4156. http://doi.org/10.1145/2556288.2557020
127. Matthew Lombard, Robert D. Reich, Maria E. Grabe, Cheryl C. Bracken, and Theresa B. Ditton. 2000. Presence and television. *Human Communication Research* 26, 1: 75–98.

128. Andrés Lucero, Jussi Holopainen, and Tero Jokela. 2011. Pass-them-around: Collaborative use of mobile phones for photo sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11), 1787–1796. http://doi.org/10.1145/1978942.1979201
129. Lumus. See the big picture. Retrieved April 26, 2016 from http://lumus-optical.com/
130. Giuliano Maciocci, Andrew J. Everitt, Paul Mabbutt, and David T. Berry. 2012. Anchoring virtual images to real world surfaces in augmented reality systems. Retrieved August 6, 2016 from http://www.google.ca/patents/US20120249741
131. Giuliano Maciocci and James Joseph Mulholland. 2013. User interface interaction for transparent head-mounted displays. Retrieved August 6, 2016 from http://www.google.ca/patents/US20130335303
132. Andrew Maimone, Douglas Lanman, Kishore Rathinavel, Kurtis Keller, David Luebke, and Henry Fuchs. 2014. Pinlight Displays: Wide field of view augmented reality eyeglasses using defocused point light sources. *ACM Trans. Graph.* 33, 4: 89:1–89:11. http://doi.org/10.1145/2601097.2601141
133. Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. 2012. Cross-device interaction via micro-mobility and F-formations. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (UIST '12), 13–22. http://doi.org/10.1145/2380116.2380121
134. Paul Marshall, Yvonne Rogers, and Nadia Pantidi. 2011. Using F-formations to analyse spatial patterns of interaction in physical environments. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work* (CSCW '11), 445–454. http://doi.org/10.1145/1958824.1958893
135. Tom Mayer, Susan Brady, Elizabeth Bovasso, Priscilla Pope, and Robert Gatchel. Noninvasive measurement of cervical tri-planar motion in normal subjects. *Spine* 18, 15: 2191-2195.
136. David C. McCallum and Pourang Irani. 2009. ARC-Pad: Absolute+relative cursor positioning for large displays with a mobile touchscreen. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (UIST '09), 153–156. http://doi.org/10.1145/1622176.1622205
137. Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. 2011. Interactive furniture layout using interior design guidelines. In *ACM SIGGRAPH 2011 Papers* (SIGGRAPH '11), 87:1–87:10. http://doi.org/10.1145/1964921.1964982
138. Meta Company. Augmented Reality. Retrieved April 20, 2016 from https://www.metavision.com/
139. Microsoft. Microsoft HoloLens. Retrieved April 20, 2016 from https://www.microsoft.com/microsoft-hololens/en-us
140. Microsoft. Kinect - Windows app development. Retrieved April 17, 2016 from https://developer.microsoft.com/en-us/windows/kinect
141. Microsoft Research. RoomAlive Toolkit. Retrieved April 17, 2016 from http://research.microsoft.com/en-us/projects/roomalivetoolkit/
142. Paul Milgram, Haruo Takemura, Akira Utsumi, and Fumio Kishino. 1995. Augmented reality: a class of displays on the reality-virtuality continuum. *Photonics*

for industrial applications. International Society for Optics and Photonics, 282–292. http://doi.org/10.1117/12.197321

143. Mark R. Mine. 1995. *Virtual Environment Interaction Techniques*. University of North Carolina at Chapel Hill, *computer science technical report TR95-018*, 507248-2, Chapel Hill, NC, USA.
144. Mark R. Mine, Frederick P. Brooks Jr, and Carlo H. Sequin. 1997. Moving objects in space: exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th Annual Conference On Computer Graphics And Interactive Techniques*, 19–26.
145. Cameron S. Miner, Denise M. Chan, and Christopher Campbell. 2001. Digital jewelry: Wearable technology for everyday life. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '01), 45–46. http://doi.org/10.1145/634067.634098
146. Arpit Mittal, Giuliano Maciocci, Michael L. Tunmer, and Paul Mabbutt. 2014. Human-body-gesture-based region and volume selection for HMD. Retrieved August 6, 2016 from http://www.google.ca/patents/US20140225918
147. David Molyneaux, Shahram Izadi, David Kim, et al. 2012. Interactive environment-aware handheld projectors for pervasive computing spaces. In *Proceedings of the 10th International Conference on Pervasive Computing* (Pervasive'12), 197–215. http://doi.org/10.1007/978-3-642-31205-2_13
148. Meredith Ringel Morris and Eric Horvitz. 2007. SearchTogether: An interface for collaborative Web search. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (UIST '07), 3–12. http://doi.org/10.1145/1294211.1294215
149. Annette Mossel, Benjamin Venditti, and Hannes Kaufmann. 2013. 3DTouch and HOMER-S: Intuitive manipulation techniques for one-handed handheld augmented reality. In *Proceedings of the Virtual Reality International Conference: Laval Virtual* (VRIC '13), 12:1–12:10. http://doi.org/10.1145/2466816.2466829
150. Miguel A. Nacenta, Satoshi Sakurai, Tokuo Yamaguchi, et al. 2007. E-conic: A perspective-aware interface for multi-display environments. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (UIST '07), 279–288. http://doi.org/10.1145/1294211.1294260
151. Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Pourang P. Irani, and Michel Beaudouin-Lafon. 2013. High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 831–840. http://doi.org/10.1145/2470654.2470773
152. NASA. 1995. *Man-Systems Integration Standards (MSIS) Rev B*. Retrieved April 18, 2016 from http://msis.jsc.nasa.gov/
153. Joseph Newman, David Ingram, and Andy Hopper. 2001. Augmented reality in a wide area sentient environment. In *Proceedings of IEEE and ACM International Symposium on Augmented Reality,* (ISMAR '01), 77–86. http://doi.org/10.1109/ISAR.2001.970517

154. Tao Ni, Doug A. Bowman, and Jian Chen. 2006. Increased display size and resolution improve task performance in information-rich virtual environments. In *Proceedings of Graphics Interface 2006* (GI '06), 139–146.
155. Bruce J. Noble, Gunnar A. Borg, Ira Jacobs, R. Ceci, and P. Kaiser. 1983. A category-ratio perceived exertion scale: relationship to blood and muscle lactates and heart rate. *Medicine and Science in Sports and Exercise* 15, 6: 523–528.
156. Oculus. Oculus Rift. Retrieved April 22, 2016 from https://www.oculus.com/en-us/rift/
157. Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. 2012. iRing: intelligent ring using infrared reflection. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12), 131–136.
158. Alex Olwal and Steven Feiner. 2003. The flexible pointer: An interaction technique for selection in augmented and virtual reality. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (UIST '03), 81–82.
159. Jason Orlosky, Kiyoshi Kiyokawa, and Haruo Takemura. 2013. Dynamic text management for see-through wearable and heads-up display systems. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces* (IUI '13), 363–370. http://doi.org/10.1145/2449396.2449443
160. Jason Orlosky, Kiyoshi Kiyokawa, and Haruo Takemura. 2014. Managing mobile text in head mounted displays: Studies on visual preference and text placement. *SIGMOBILE Mob. Comput. Commun. Rev.* 18, 2: 20–31. http://doi.org/10.1145/2636242.2636246
161. Robert Patterson, Marc D. Winterbottom, and Byron J. Pierce. 2006. Perceptual issues in the use of head-mounted visual displays. *Human Factors* 48, 3: 555–573.
162. Thammathip Piumsomboon, David Altimira, Heonhwan Kim, Andrew Clark, Gun Lee, and Mark Billinghurst. 2014. Grasp-shell vs gesture-speech: A comparison of direct and indirect natural interaction techniques in augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality* (ISMAR '14), 73–82.
163. Catherine Plaisant, David Carr, and Ben Shneiderman. 1995. Image-browser taxonomy and guidelines for designers. *IEEE Softw.* 12, 2: 21–32. http://doi.org/10.1109/52.368260
164. Adam G. Poulos, Evan Michael Keibler, Arthur Tomlin, et al. 2016. Interactions of virtual objects with surfaces. Retrieved August 6, 2016 from http://www.google.ca/patents/US9245388
165. Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The Go-go interaction technique: Non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology* (UIST '96), 79–80. http://doi.org/10.1145/237091.237102
166. Ivan Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. 1998. Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In *Computer graphics forum*, 17, 3. Blackwell Publishers Ltd., 41–52.
167. Zachary Pousman and John Stasko. 2006. A taxonomy of ambient information systems: four patterns of design. In *Proceedings Of The Working Conference On Advanced Visual Interfaces*, 67–74.

168. Zachary Pousman, John T. Stasko, and Michael Mateas. 2007. Casual information visualization: Depictions of data in everyday life. *Visualization and Computer Graphics, IEEE Transactions on* 13, 6: 1145–1152.
169. Jérôme Prado, Simon Clavagnier, Hélène Otzenberger, Christian Scheiber, Henry Kennedy, and Marie-Thérèse Perenin. 2005. Two cortical systems for reaching in central and peripheral vision. *Neuron* 48, 5: 849–858. http://doi.org/10.1016/j.neuron.2005.10.010
170. Blaine A. Price, Ronald M. Baecker, and Ian S. Small. 1993. A principled taxonomy of software visualization. *Journal of Visual Languages & Computing* 4, 3: 211–266. http://doi.org/10.1006/jvlc.1993.1015
171. Umar Rashid, Miguel A. Nacenta, and Aaron Quigley. 2012. The cost of display switching: A comparison of mobile, large display and hybrid ui configurations. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (AVI '12), 99–106. http://doi.org/10.1145/2254556.2254577
172. Ramesh Raskar, Jeroen van Baar, Paul Beardsley, Thomas Willwacher, Srinivas Rao, and Clifton Forlines. 2003. iLamps: Geometrically aware and self-configuring projectors. In *ACM SIGGRAPH 2003 Papers* (SIGGRAPH '03), 809–818. http://doi.org/10.1145/1201775.882349
173. Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. 1998. The Office of the Future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (SIGGRAPH '98), 179–188. http://doi.org/10.1145/280814.280861
174. Ramesh Raskar, Greg Welch, and Henry Fuchs. 1999. Spatially augmented reality. In *Proceedings of the International Workshop on Augmented Reality : Placing Artificial Objects in Real Scenes* (IWAR '98), 63–72.
175. Ramesh Raskar, Greg Welch, Kok-Lim Low, and Deepak Bandyopadhyay. 2001. Shader Lamps: Animating real objects with image-based illumination. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, 89–102.
176. Jun Rekimoto and Katashi Nagao. 1995. The world through the computer: Computer augmented interaction with real world environments. In *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology* (UIST '95), 29–36. http://doi.org/10.1145/215585.215639
177. Jun Rekimoto and Masanori Saitoh. 1999. Augmented surfaces: A spatially continuous work space for hybrid computing environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '99), 378–385. http://doi.org/10.1145/302979.303113
178. George Robertson, Maarten van Dantzich, Daniel Robbins, et al. 2000. The Task Gallery: A 3D window manager. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '00), 494–501. http://doi.org/10.1145/332040.332482
179. Warren Robinett. 1992. *Synthetic Experience: A Taxonomy, Survey of Earlier Thought, and Speculations on the Future*. University of North Carolina at Chapel Hill, Chapel Hill, NC, USA.

180. Warren Robinett. 1994. Synthetic experience. In *State of the Art in Computer Graphics*. Springer, 177–194.
181. Renato F. Salas-Moreno, Ben Glocken, Paul H. J. Kelly, and Andrew J. Davison. 2014. Dense planar SLAM. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 157–164. http://doi.org/10.1109/ISMAR.2014.6948422
182. Joey Scarr, Andy Cockburn, Carl Gutwin, and Andrea Bunt. 2012. Improving command selection with CommandMaps. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), 257–266. http://doi.org/10.1145/2207676.2207713
183. Dieter Schmalstieg, L. Miguel Encarnação, and Zsolt Szalavári. 1999. Using transparent props for interaction with the virtual table. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics* (I3D '99), 147–153. http://doi.org/10.1145/300523.300542
184. Takashi Shibata, Joohwan Kim, David M. Hoffman, and Martin S. Banks. 2011. The zone of comfort: Predicting visual discomfort with stereo displays. *Journal of Vision* 11, 8: 11. http://doi.org/10.1167/11.8.11
185. Ben Shneiderman. 1983. Direct manipulation: A step beyond programming languages. *Computer* 16, 8: 57–69. http://doi.org/10.1109/MC.1983.1654471
186. Lauren Shupp, Robert Ball, Beth Yost, John Booker, and Chris North. 2006. Evaluation of viewport size and curvature of large, high-resolution displays. In *Proceedings of Graphics Interface 2006* (GI '06), 123–130.
187. Nathan Silberman, Lior Shapira, Ran Gal, and Pushmeet Kohli. 2014. A Contour Completion model for augmenting surface reconstructions. In *Computer Vision – ECCV 2014*. Springer International Publishing, 488–503.
188. Hyunyoung Song, Tovi Grossman, George Fitzmaurice, et al. 2009. PenLight: Combining a mobile projector and a digital pen for dynamic visual overlay. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09), 143–152. http://doi.org/10.1145/1518701.1518726
189. Hyunyoung Song, Francois Guimbretiere, Tovi Grossman, and George Fitzmaurice. 2010. MouseLight: Bimanual interactions on digital paper using a pen and a spatially-aware mobile projector. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 2451–2460. http://doi.org/10.1145/1753326.1753697
190. Mark St. John, Wayne C. Harris, and Glenn Osga. 1997. Designing for multi-tasking environments: multiple monitors vs. multiple windows. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 41, 2: 1313–1317. http://doi.org/10.1177/107118139704100213O
191. John Stasko, Carsten Gorg, Zhicheng Liu, and Kanupriya Singhal. 2007. Jigsaw: Supporting investigative analysis through interactive visualization. In *IEEE Symposium on Visual Analytics Science and Technology, 2007* (VAST 2007), 131–138. http://doi.org/10.1109/VAST.2007.4389006
192. Markus Steinberger, Manuela Waldner, Marc Streit, Alexander Lex, and Dieter Schmalstieg. 2011. Context-preserving visual links. *IEEE Transactions on Visualization and Computer Graphics,* 17, 12: 2249–2258.

193. Richard Stoakley, Matthew J. Conway, and Randy Pausch. 1995. Virtual Reality on a WIM: Interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '95), 265–272. http://doi.org/10.1145/223904.223938
194. David J. Sturman and David Zeltzer. 1994. A survey of glove-based input. *IEEE Computer Graphics and Applications* 14, 1: 30–39. http://doi.org/10.1109/38.250916
195. Ramona E. Su and Brian P. Bailey. 2005. Put them where? Towards guidelines for positioning large displays in interactive workspaces. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction* (INTERACT'05), 337–349. http://doi.org/10.1007/11555261_29
196. Ivan E. Sutherland. 1965. The ultimate display. In *Proceedings of the IFIP Congress*, 506–508.
197. Ivan E. Sutherland. 1968. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, 757–764.
198. Z. Szalavári, D. Schmalstieg, A. Fuhrmann, and M. Gervautz. 1998. "Studierstube": An environment for collaboration in augmented reality. *Virtual Reality* 3, 1: 37–48. http://doi.org/10.1007/BF01409796
199. Zsolt Szalavári and Michael Gervautz. 1997. The Personal interaction panel – a Two-handed interface for augmented reality. *Computer Graphics Forum* 16, 3: C335–C346. http://doi.org/10.1111/1467-8659.00137
200. Seiichiro Tabata. 1996. Image display apparatus of head mounted type. Retrieved August 6, 2016 from http://www.google.ca/patents/US5579026
201. Susanne Tak, Andy Cockburn, Keith Humm, David Ahlström, Carl Gutwin, and Joey Scarr. 2009. Improving window switching interfaces. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II* (INTERACT '09), 187–200. http://doi.org/10.1007/978-3-642-03658-3_25
202. Desney Tan and Mary Czerwinski. 2003. Effects of visual separation and physical discontinuities when distributing information across multiple displays. In *Human-computer interaction, INTERACT '03 : IFIP TC13 International Conference on Human-Computer Interaction* (INTERACT '03), 252-255.
203. Kohei Tanaka, Yasue Kishino, Masakazu Miyamae, Tsutomu Terada, and Shojiro Nishio. 2008. 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 '08), 139–142. http://doi.org/10.1109/ISMAR.2008.4637340
204. Feng Tang, Ang Li, and Xiaojin Shi. 2016. Three-dimensional hand tracking using depth sequences. Retrieved August 6, 2016 from http://www.google.ca/patents/WO2016025713A1
205. Robert J. Teather and Wolfgang Stuerzlinger. 2011. Pointing at 3D targets in a stereo head-tracked virtual environment. In *Proceedings of the 2011 IEEE Symposium on 3D User Interfaces* (3DUI '11), 87–94.
206. Alexander Toet, Sander E. M. Jansen, and Nico J. Delleman. 2007. Effects of field-of-view restrictions on speed and accuracy of manoeuvring. *Perceptual and Motor Skills* 105, 3 Pt 2: 1245–1256. http://doi.org/10.2466/pms.105.4.1245-1256

207. Tom Ritchey. 1998. Fritz Zwicky, morphologie and policy analysis. In *16th EURO conference on operational analysis, Brussels*.
208. Anne M. Treisman and Garry Gelade. 1980. A feature-integration theory of attention. *Cognitive Psychology* 12, 1: 97–136. http://doi.org/10.1016/0010-0285(80)90005-5
209. John Underkoffler, Brygg Ullmer, and Hiroshi Ishii. 1999. Emancipated Pixels: Real-world Graphics in the Luminous Room. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (SIGGRAPH '99), 385–392. http://doi.org/10.1145/311535.311593
210. Dimitar Valkov, Frank Steinicke, Gerd Bruder, et al. 2010. Touching floating objects in projection-based virtual reality environments. In *Proceedings of the 16th Eurographics Conference on Virtual Environments & Second Joint Virtual Reality* (EGVE - JVRC'10), 17–24. http://doi.org/10.2312/EGVE/JVRC10/017-024
211. Lode Vanacken, Tovi Grossman, and Karin Coninx. 2007. Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. In *2007 IEEE Symposium on 3D User Interfaces* (3DUI '07). http://doi.org/10.1109/3DUI.2007.340783
212. Frederic Vernier and Laurence Nigay. 2000. A framework for the combination and characterization of output modalities. In *International Workshop on Design, Specification, and Verification of Interactive Systems*, 35–50.
213. Harry Veron, P. J. Hezel, and David A. Southard. 1994. Head-mounted displays for virtual reality. In *SPIE's International Symposium on Optical Engineering and Photonics in Aerospace Sensing*. International Society for Optics and Photonics, 41–50. http://doi.org/10.1117/12.177384
214. John Vilk, David Molnar, Benjamin Livshits, et al. 2015. SurroundWeb: Mitigating privacy concerns in a 3D Web browser. In *2015 IEEE Symposium on Security and Privacy*, 431–446. http://doi.org/10.1109/SP.2015.33
215. John Vilk, David Molnar, Eyal Ofek, et al. 2014. *Least Privilege Rendering in a 3D Web Browser*. Technical report, Microsoft Research.
216. Daniel Vogel and Ravin Balakrishnan. 2005. Distant freehand pointing and clicking on very large, high resolution displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (UIST '05), 33–42. http://doi.org/10.1145/1095034.1095041
217. Vuforia. Retrieved April 17, 2016 from http://www.vuforia.com/
218. Daniel Wagner and Dieter Schmalstieg. 2006. Handheld augmented reality displays. In *IEEE Virtual Reality Conference* (VR 2006), 321–321. http://doi.org/10.1109/VR.2006.67
219. Manuela Waldner, Raphael Grasset, Markus Steinberger, and Dieter Schmalstieg. 2011. Display-adaptive window management for irregular surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (ITS '11), 222–231. http://doi.org/10.1145/2076354.2076394
220. Manuela Waldner, Werner Puff, Alexander Lex, Marc Streit, and Dieter Schmalstieg. 2010. Visual links across applications. In *Proceedings of Graphics Interface 2010* (GI

’10), 129–136. Retrieved April 16, 2016 from http://dl.acm.org/citation.cfm?id=1839214.1839238
221. QianYing Wang, Tony Hsieh, and Andreas Paepcke. 2009. Piles across space: Breaking the real-estate barrier on small-display devices. *Int. J. Hum.-Comput. Stud.* 67, 4: 349–365. http://doi.org/10.1016/j.ijhcs.2008.10.004
222. Rui Wang, Fanglin Chen, Zhenyu Chen, et al. 2014. StudentLife: Assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (UbiComp ’14), 3–14. http://doi.org/10.1145/2632048.2632054
223. Mathias Wilhelm, Daniel Krakowczyk, Frank Trollmann, and Sahin Albayrak. 2015. eRing: multiple finger gesture recognition with one ring using an electric field. In *Proceedings of the 2nd international Workshop on Sensor-based Activity Recognition and Interaction*. ACM, 1–6. http://doi.org/10.1145/2790044.2790047
224. Peter Willemsen, Mark B. Colton, Sarah H. Creem-Regehr, and William B. Thompson. 2009. The effects of head-mounted display mechanical properties and field of view on distance judgments in virtual environments. *ACM Trans. Appl. Percept.* 6, 2: 8:1–8:14. http://doi.org/10.1145/1498700.1498702
225. Andrew D. Wilson. 2005. PlayAnywhere: A compact interactive tabletop projection-vision system. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (UIST ’05), 83–92. http://doi.org/10.1145/1095034.1095047
226. Andrew D. Wilson and Hrvoje Benko. 2010. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (UIST ’10), 273–282.
227. Graham Wilson, Thomas Carter, Sriram Subramanian, and Stephen A. Brewster. 2014. Perception of ultrasonic haptic feedback on the hand: Localisation and apparent motion. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (CHI ’14), 1133–1142. http://doi.org/10.1145/2556288.2557033
228. Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2012. Magic Finger: Always-available input through finger instrumentation. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (UIST ’12), 147–156. http://doi.org/10.1145/2380116.2380137
229. Xing-Dong Yang, Edward Mak, David McCallum, Pourang Irani, Xiang Cao, and Shahram Izadi. 2010. LensMouse: Augmenting the mouse with an interactive touch display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’10), 2431–2440. http://doi.org/10.1145/1753326.1753695
230. Ka-Ping Yee. 2003. Peephole Displays: Pen interaction on spatially aware handheld computers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’03), 1–8. http://doi.org/10.1145/642611.642613
231. Ji Soo Yi, Youn ah Kang, John T. Stasko, and Julie A. Jacko. 2007. Toward a deeper understanding of the role of interaction in information visualization. In *IEEE Transactions on Visualization and Computer Graphics,* 13, 6: 1224–1231.

232. Sang Ho Yoon, Ke Huo, and Karthik Ramani. 2014. Plex: finger-worn textile sensor for mobile interaction during activities. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, ACM, 191–194. http://doi.org/10.1145/2638728.2638746
233. Suya You, U. Neumann, and R. Azuma. 1999. Hybrid inertial and vision tracking for augmented reality registration. In *Proceedings of IEEE Virtual Reality*, 260–267. http://doi.org/10.1109/VR.1999.756960
234. Shumin Zhai, William Buxton, and Paul Milgram. 1994. The “Silk Cursor”: Investigating transparency for 3d target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’94), 459–464. http://doi.org/10.1145/191666.191822