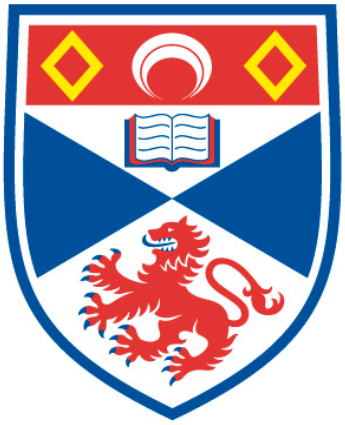
DESIGNING SPATIALLY-AWARE INDOOR VISUAL INTERFACES AND SYSTEMS

Jakub Dostál

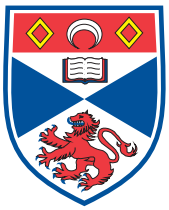
A Thesis Submitted for the Degree of PhD
at the
University of St Andrews



2016

Designing Spatially-Aware Indoor Visual Interfaces and Systems

Jakub Dostál



University
of
St Andrews

This thesis is submitted in partial fulfilment for the degree of

PhD

at the

University of St Andrews

April 2016

Abstract

The environments in which people interact with displays and other devices are changing. Interactions are not longer constrained by displays being tethered to a desk. As the variety and complexity of interactive environments increases, so does the importance of spatial aspects of interactions and the physical and visual constraints of people and other interactive entities.

This thesis examines spatial relationships between entities and other characteristics of interactions through the lens of the Interaction Relationship Entity model, also introduced here.

Moreover, the thesis demonstrates the viability of low-cost, high-availability hardware and software for exploration of novel interactive systems through a set of algorithms that can be used for spatial tracking.

The presented work also includes three case studies, each of which explores different aspects of spatial interactions with displays. The first case study investigates the use of displays capable of simultaneously showing two different views from different angles for creating spatial interactions that do not require active tracking. The second case study explores dynamic manipulation of on-display content and prototyping spatial interactions with large displays. The third case study considers how visual changes on displays in a multi-display environment can be tracked during periods of inattention.

Candidate's Declaration

I, Jakub Dostál, hereby certify that this thesis, which is approximately 75,500 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in August 2010, and as a candidate for the degree of PhD in August 2010; the higher study for which this is a record was carried out in the University of St Andrews between 2010 and 2015.

Date ..... Signature of Candidate .....

Supervisor's Declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of PhD in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date ..... Signature of Supervisor .....

Permission for publication

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. I have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

Embargo for a period of 1 year on the electronic and print copy as their publication would preclude future publication, for which the content is being prepared.

Date ..... Signature of Candidate .....

Signature of Supervisor .....

Acknowledgements

Completing this thesis would not have been possible without the support of many others. First of all, I would like to thank my supervisors, Prof. Aaron Quigley and Dr. Per Ola Kristensson for their guidance and support throughout my PhD studies. Being able to learn from their examples has allowed me to grow both as a researcher and as a person.

Great thanks also goes to all of the collaborators, with whom I have been fortunate to work. Not all the collaborations have had direct influence on the content presented in this thesis but in all cases, the collaborations have been informative to aspects of my research — thank you Uta Hinrichs, Saul Greenberg, David Kim, Jo vermeulen, Nic Marquardt, Shahram Izadi, Christoph Rhemann, Cem Ceskin, Jamie Shotton, and others at Microsoft Research.

I would also like to thank all the members of the School of Computer Science at the University of St Andrews and especially the SACHI group for helping create an amazing place for work and play.

I am very grateful to SICSA for providing financial support for parts of the research presented here, as well as creating many unique opportunities for me and others.

Last, but most definitely not least, I would like to thank my parents and Lizzie, for their unconditional, never-ending support and patience.

Data Management

Where data has been collected during any evaluation, case study or experiment, the data is being managed according to the permissions agreed to by the participants through consent forms. This generally means that collected data is only available to the researchers directly involved in the evaluation due to the potentially private and sensitive nature of the collected data (most commonly images of people's likeness). The results of and findings from the evaluations are presented in the relevant parts of this thesis.

In accordance with the participants' consent, all data collected from the participants is stored securely within a locked office and will be deleted or safely disposed of within five (5) years of collection. This includes collected images, questionnaire and answer sheets or any other data provided by the participants.

Systems developed for this thesis or code used to process data from evaluations are available as source code or binary files as appropriate on request to the School of Computer Science.

Supplementary video material is available on the attached DVD, or included in an on-line appendix at `http://dostal.co.uk/phd`.

Contents

Chapter 1

Introduction

In 1991, Mark Weiser coined the term *ubiquitous computing* to describe the concept of interactive systems that become a part of the fabric of our everyday lives [Wei91]. Arguably, computers are much more ubiquitous and pervasive in our everyday lives now then they were at the time Mark Weiser coined the term, even if they do not always quite "fade into the background".

We have moved from an era of personal desktop computers, usually tethered to desks, into a period where mobile devices such as smartphones and tablets can be the main form of a person's computing interactions. At the same time, on the opposite end of the size spectrum, large displays in various forms are becoming more ubiquitous and varied. Together with advancements in sensing technologies (e.g. the Microsoft Kinect sensor), these have increased the scope for interaction enormously by increasing access to commodity spatial tracking. Spatial interactions, one of the focal points of this thesis, in particular benefit from these advances.

However, even before Weiser's article there had been researchers exploring spatial interactions, although often under the umbrella of context aware or responsive environments. One of the first examples was a set of responsive environments created by Krueger et al. [Kru77; KGH85]. Some of these used an augmented floor, which allowed the positions and movements of a person within a room to be tracked.

By the time of Weiser's introduction of the ubiquitous computing concept, Roy and Want were already working with a system called Active Badge [WH92; WH+92], which was a building-scale spatially-aware system that tracked people using badges augmented with infrared emitters.

In 1993, Fitzmaurice [Fit93] published work on situated information spaces and spatially-aware handheld computers. In the same year, a spatial model of interaction aimed primarily at virtual environments was published by Benford et. al. [Ben93; BB+93]. This model became known as the Focus/Nimbus model and would become influential for physical spatial interactions as well as virtual ones. The model was later extended by Rodden [Rod96] to further extend its potential uses. In approximately the same period, the concept of context-aware computing was being introduced by Schilit et al. [SAW94; Sch95].

In the following years, a number of other spatially-aware systems were explored by researchers. These included the CyberGuide system [AA+96], the CyberDesk [Dey98], the EasyLiving project [SKB98] or the work on augmented surfaces by Rekimoto et al. [RS99]. While most of the systems focused on indoor environments, outdoor environments were also explored by researchers (e.g. [CM+00]).

In addition to research systems exploring various novel interactions, a number of more theoretical contributions were published after the year 2000. Dey [Dey00; DAS01] created a conceptual framework and a toolkit for prototyping context-aware applications. Dix et al. [DR+00] introduced a design framework for interactive mobile systems.

The number of systems making use of spatial interactions has only increased since then and a large number of them are examined at various points in this thesis. While spatial interactions are slowly becoming more common in our everyday lives and not just as part of research explorations, much remains to be done to fully understand their potential. This thesis aims to increase our understanding of the use of spatial interactions, their strengths and weaknesses as well as providing a more structured view of existing work.

1.1 Scope

The primary focus of this thesis is on indoor environments. While outdoor systems have also been explored in previous research, this is out of the scope of the work presented here. This is because the variety of contexts of use is very broad even when constrained only to indoor environments. The research presented in this thesis includes explorations of spatial interactions ranging from small-scale desktop scenarios to room-level interactions. Existing systems analysed in this thesis use an even greater range of size of interactive environments (from personal interactions with a mobile phone to building-scale interactive systems).

This thesis focuses on the use of spatial relationship for interactions in terms of the spatial arrangements between entities. The social aspects of the interactions such as collaboration, sharing, territoriality, or conflict resolution are mostly outside of the scope of the presented research. Several researchers have already explored some of the more social aspects of interactions. For example, Scott [SSI04; Sco05] has explored territoriality in collaborative tabletop interactions. More recently Marquardt et al. [Mar13; BMG10; GM+11; MDM+11] applied the concept of proxemics by Edward Hall [Hal66] to interactive systems. More generally, research in the area of Computer-Supported Collaborative Work[1] is likely to be relevant to a reader interested in the social aspects of interactions.

1.2 Research Questions

The central hypothesis of this thesis is that *studying how spatial relationships can be leveraged in indoor environments for interactive purposes can enable development of novel interactions*.

The hypothesis is addressed through the following research activities: an analysis of the use of spatial relationships in existing systems within a conceptual model; the development and implementation of an experimental platform for prototyping of spatially aware systems; and an exploration of the design space for interfaces using spatial relationships for interactions. These activities define the following research research questions, which will be answered in this thesis:

Question 1: How can the use of spatial relationships in existing interactive systems be analysed? In order to answer this question, I designed a conceptual model for the analysis of spatial relationships in interactive systems, with a particular focus on the types of interactive entities and the spatial relationships between them. Additionally, the newly designed conceptual model is also linked to a number of existing models, frameworks and other conceptual constructs in order to create an even richer set of resources that can be used for analysis.

Question 2: How are spatial relationships currently used by existing indoor systems? To answer this question, I analysed a number of existing research within the context of the conceptual model created as part of the answer to *Question 1*.

[1] See e.g. [Gre91; Bae93] and the proceedings of the ACM CSCW conference for starting points.

Question 3: How can the prototyping of spatially aware indoor visual interfaces be supported? This question is answered in several ways. Firstly, I designed and implemented several computer vision-based tracking algorithms. Secondly, I also developed a toolkit to facilitate faster prototyping of spatially-aware visual interfaces.

Question 4: Can the prototyping support tools created in this thesis be used to create novel interactive systems? This question is answered through the creation of four different prototype systems, each of which has a different focus. Two of the prototypes demonstrate that it is possible to design systems that enable spatial interactions that do not always require active tracking. Another prototype concentrates on the use of on-display content manipulation. The last prototype shows how indirect use of spatial relationships can be leveraged to create novel interactions.

The research presented in this thesis will likely benefit three audiences. Researchers and practitioners can use the conceptual model used to answer *Research Question 1* to find comparable systems, to locate gaps in existing research and possibly also to analyse requirements during while designing a new system. System designers and researches may use the prototyping tools introduces in this thesis to support the creation of early versions of their new systems. Lastly, readers looking to deepen their understanding of spatial interactions will likely benefit from reading this thesis.

1.3 Contributions

This thesis makes a number of interconnected contributions. Three main contributions are summarised here. The thesis introduces the Interaction Relationship Entity (IRE) model, which focusses on the spatial relationships between interactive entities, while also taking into account more general interaction characteristics. The model and its analytical application to existing literature form the first main contribution. The second main contribution are a set of tracking algorithms, which primarily leverage common RGB cameras to create a markerless tracking system, without the need for highly specialised or expensive hardware. The third main contribution is a set of three case studies that explore different ways, in which displays and their properties can be utilised for spatial interactions.

Additionally several smaller contributions are made throughout the thesis. These include both theoretical and design contributions. The most significant of the minor contributions include a set of four subtle visualisation techniques for tracking change on unattended displays and a toolkit for rapid prototyping of dynamic manipulation of visual content on displays.

1.4 Publications

Some of the ideas presented in this thesis have already been published in one of the following publications, while other parts of the thesis were motivated of influenced by the publications. Where co-authored published content is part of the thesis, only work authored and/or carried out by the author of this thesis is included, unless explicitly acknowledged otherwise.

Journal Articles

(J.1) Estimating and using absolute and relative viewing distance in interactive systems. Jakub Dostal, Per Ola Kristensson and Aaron Quigley. In *Journal of Pervasive and Mobile Computing*, Volume 10, February, 2014. pages 173–186, doi:10.1016/j.pmcj.2012.06.009.
Content from this article is included in Chapter 4.

(J.2) The Dark Patterns of Proxemic Sensing. Sebastian Boring, Saul Greenberg, Jo Vermeulen, Jakub Dostal, Nicolai Marquardt. In *IEEE Computer*, Volume 47, Issue 8, August, 2014. pages 56-60,

doi:10.1109/MC.2014.223.
No content from this article has been included in this thesis, however, the article is referenced in Chapter 2 and helped motivate the sampling of the design space.

Full Conference Papers

(C.1) Subtle Gaze-Dependent Techniques for Visualising Display Changes in Multi-Display Environments. Jakub Dostal, Per Ola Kristensson and Aaron Quigley. In *Proceedings of the 18th ACM International Conference on Intelligent User Interfaces (IUI 2013)*, pages 137–148 ACM, 2013.
Content from this paper appears primarily in Chapter 7. Additionally, content related to the tracking system is included in Chapter 4.

(C.2) Multi-View Proxemics: Distance and Position Sensitive Interaction. Jakub Dostal, Per Ola Kristensson and Aaron Quigley. In *Proceedings of the 2nd International Symposium on Pervasive Displays (PerDis 2013)*, pages 1–6, ACM, 2013.
Content from this paper appears in Chapter 5, with implementation details about the tracking system included in Chapter 4.

(C.3) SpiderEyes: Designing Attention and Proximity-Aware Collaborative Interfaces for Wall-Sized Displays. Jakub Dostal, Uta Hinrichs, Per Ola Kristensson and Aaron Quigley. In *Proceedings of the 19th ACM International Conference on Intelligent User Interfaces (IUI 2014)*, pages 143-152, ACM, 2014.
Content from this paper appears primarily in Chapter 6, with details about the tracking system included in Chapter 4.

(C.5) Dark Patterns in Proxemic Interactions: A Critical Perspective. Saul Greenberg, Sebastian Boring, Jo Vermeulen, Jakub Dostal. In *Proceedings of the 10th ACM Conference on Designing Interactive Systems (DIS 2014)*, pages ACM, 2014. (Best Paper Award)
No content from this article has been included in this thesis. However, the paper is referenced in Chapter 2.

Workshop Papers and Other Peer-reviewed Publications

(W.1) The Potential of Fusing Computer Vision and Depth Sensing for Accurate Distance Estimation. Jakub Dostal, Per Ola Kristensson and Aaron Quigley. In *Extended Abstracts of the 31st ACM Conference on Human Factors in Computing Systems (CHI 2013) (Work-In-Progress)*, pages 1257–1262, ACM, 2013.
Content from this paper is included in Chapter 4.

(W.2) Designing Mobile Computer Vision Applications for the Wild: Implications on Design and Intelligibility. Per Ola Kristensson, Jakub Dostal and Aaron Quigley. In *Pervasive Intelligibility: the Second Workshop on Intelligibility and Control in Pervasive Computing*, 2012.
Content from this paper is included in Chapter 4.

1.5 Thesis Structure

This thesis can be divided into three main sections — conceptual and research context; technological support platform; and results and implications of design space explorations. In the first section, the conceptual views in this thesis are framed through the introduction of an interaction model. The

application of the model provides an overview of related work in terms of interactive systems, as well as revealing trends and gaps in related research. Other related literature is placed close to where it is used, helping to contextualise the presented work.

The second main section of this thesis summarises the technological platform used to conduct experiments, as well as highlighting its constraints and the lessons learnt in its design and development. The last part of this thesis presents the results of a number of case studies, which represent a sampling of the design space for interactions that leverage both spatial relationships, and the visual and physical properties of people and displays.

The following list provides a quick overview of the structure of the remainder of this thesis:

Chapter 2 - The Interaction Relationship Entity Model — This chapter helps answer *Research Question 1* by introducing a conceptual model that enables analysis and comparison of existing systems. The model also serves to conceptually frame the spatial interfaces considered in this thesis.

Chapter 3 - Existing Systems through the Lens of the IRE Model — The analysis in this chapter addresses *Research Question 2* by using the conceptual model introduced in the preceding chapter to analyse and compare existing spatially-aware indoor systems. Additionally, the analysis positions the case studies presented later in the thesis in the larger context of the field, deepens the understanding of existing research into spatial interfaces and highlights gaps in previous research.

Chapter 4 - Computer Vision Tracking Technologies for Spatially-Aware Interfaces — This chapter presents the technological platform enabling the creation of the prototype systems in this thesis. The technological platform also in part addresses *Research Question 3*. The main contributions of this chapter are the tracking algorithms that enable low-cost markerless tracking of single and multiple individuals. The evaluations of the algorithms and some of their underlying features are also included in this chapter.

Chapter 5 - Case Study 1 - Interaction with a Multi-View Display — The first case study, *MultiView*, presents an investigation of utilisation of inherent properties of objects (specifically displays) to create spatially interactive interfaces without need for extensive tracking and dynamic content manipulation. This case study highlights that there are other avenues to spatial interfaces than resource intensive tracking. It also forms part of the answer to *Research Question 4*.

Chapter 6 - Case Study 2 - Prototyping Multi-User Interactions with a Large Display — The second case study, *SpiderEyes*, presents a systematic enquiry into the breath of possibilities in dynamic visual manipulation of on-display content on large displays. In addition, the chapter also contributes a rapid prototyping tool as well as a set of observations and recommendations. The contributions in this chapter help address *Research Questions 3* and *4*.

Chapter 7 - Case Study 3 - Techniques for Inattention in Multi-Display Environments — The last case study, *DiffDisplays*, focuses on exploring an application area for spatial interfaces that concentrates on the inverse of what most visual systems concentrate on — visual interactions that occur, while a person is *not* looking at a display. This chapter introduces four visualisation techniques for managing visual content changes on unattended display, including an evaluation of their effectiveness. The research prototype presented in this chapter helps answer *Research Question 4*.

Chapter 8 - Summary and Conclusions — This chapter summarises the main findings in this thesis and briefly explores some of their implications for future research.

Appendix - Visual Considerations for Spatially-Aware Systems — This appendix summarises a subset of literature on human vision relevant to designers of spatially-aware visual interfaces. This literature summary is augmented with additional simplified measures to support existing approaches in case of limited resources being available. Additionally, the appendix provides a set of considerations that should be taken into account when creating new systems that may be affected by the physiological and technological limitations related to spatially-aware visual interfaces.

Chapter 2

The Interaction Relationship Entity Model

This chapter introduces the Interaction Relationship (IRE) model. As the name suggests, the model centres around three concepts, entities, their relationships and other interaction characteristics. The focus of the model is on spatial relationships between entities and a set of five interaction characteristics. The spatial relationships are divided into three dimensions — distance, orientation, and position. Five interaction characteristics are included in the model — interaction range, mode, action intentionality, action intensity and entity cardinality. The chapter concludes with a brief discussion on the analytical use of the model and by examining how the model could be used in combination with other interaction models and taxonomies.

When describing his Instrumental Interaction Model, Beaudouin-Lafon [BL00] proposed three desirable characteristics of interaction models for post-WIMP[1] interfaces:

Descriptive: The model should be able to cover both existing and new applications.

Comparative: The model should provide metrics for comparing alternative designs as opposed to being *prescriptive*, deciding a priori what is good and what is bad.

Generative: The model should facilitate creation of new interaction techniques.

The Interaction Relationship Entity (IRE) model introduced here is designed to exhibit all three of the characteristics. The primary purpose of the Interaction Relationship Entity (IRE) model is to offer an analysis platform for existing research and commercial systems that make use of spatial relationships for interaction purposes. The model's descriptive abilities are demonstrated in Chapter 3, where it is used to analyse a number existing systems as well as several novel prototypes produced for this thesis.

The comparative capacity is also demonstrated in the same chapter. When the analysed systems are examined through the lens of the IRE model, it becomes possible to compare very diverse prototypes. More importantly, the resulting analysis is not prescriptive in nature, it only reveals opportunities for future exploration. The IRE model is generative in that it enables designers to first discover, and then fill, gaps with novel systems that take advantage of previously unused relationships and other interaction characteristics. By using the IRE model as a lens during design, one can clarify requirements and identify potential challenges that the system designer may face.

2.1 Entities

There are three types of entities within the IRE model - *Objects*, *Actors*, and *Environments*. The entity classes are primarily defined by the role each entity plays in the interaction scenario. The focus is on

[1] WIMP: Windows, Icons, Menus, Pointer

whether the entity is actively in control of the interaction or whether it is being interacted on, and how the entity provides context or limits to the interaction scenario.

In the remainder of the thesis, the following notation will be used when describing IRE model entities. When referring to an IRE entity, the first letter will be capitalised (Object, Actor, Environment). When the term is not capitalised, it refers to a general meaning of the word.

2.1.1 Objects

Unlike other interactional models, the IRE model does not make a distinction between devices and objects. This is primarily due to the assumption that any object can be used within an interaction scenario as either an input or an output, or both. A non-instrumented object can be used as an input through the use of external tracking and almost any object can be turned into a display surface through light projection. There are exceptions to the assumption that any object can be augmented with sensors, but these are likely to be due to practical constraints. Moreover, even objects with embedded sensors are limited in their use without additional augmentation as they are constrained to the embedded sensors. For the purpose of this thesis, objects are also defined as having a physical presence. While it is possible to use the IRE model within a virtual reality environment with minimal modification, the primary purpose of the model is for analysis of systems with physical elements.

2.1.2 Actors

An Actor is an active participant in an interaction that exerts some degree or control over it. Most commonly this would correspond to a person, which is how the category is used within this thesis. However, the primary purpose of this category is to distinguish between passively interactive *Objects* (which require external control to be used for interaction), and *Actors* (which exert active control over Objects and other entities to drive the interaction). For example, a self-actuated autonomous robot, or a pet cat could be Actors within a specific interaction scenario. On the other hand a person can be stationary and instrumented with a display (or an image can be projected on them) and if they have no control over the display and they do not form an active part of the interaction scenario, they would be considered to be an Object within said scenario.

2.1.3 Environments

An Environment describes the space in which the interactions take place. Its boundary is defined as the point(s) beyond which interaction is no longer possible. In the model, the size and shape of an Environment are constrained by two factors, the connectivity of interactional inputs and the ability to perceive the interactional outputs.

The connectivity of inputs is understood as the ability of a person to use a particular modality as an input into the interactive system. For example, for an interactive device with an embedded keyboard, the constraint is the physical distance of a person attempting to use the device to the interactive device. This is because without being able to reach the keyboard, the person cannot use it as an input. For a near-identical device, also with a keyboard input, but with the keyboard connected over Bluetooth, the constraint becomes the distance at which the Bluetooth signal is strong enough to successfully transmit to the device.

The ability to perceive interactional outputs is treated analogously. For example, in a scenario where a person is interacting with a large physical display, such as a television set, the environment is constrained to the areas where the display can be seen from. This typically means that the space behind the display is not part of the environment since the display cannot be seen from that position. The input/output distinction makes defining the extent of an Environment somewhat complex as with some systems it may be possible to still provide input when the output of the system is no

longer perceived and vice versa. Additionally, in multi-modal systems it is also possible that different modalities will have different (overlapping or distinct) ranges of availability and effectiveness.

For the purpose of this thesis, the Environment is primarily used as the union of all the possible input and output interaction areas. The extent of the environment is limited by the availability of at least one input or output modality. However, readers considering using the IRE model may wish to consider whether a more granular analysis may be more beneficial in their use case, focusing only on specific modalities, inputs or outputs.

2.1.4 Discussion

It is clear from the entity definitions that it is necessary to carefully consider the dynamics of the interaction scenario that is being analysed. One should consider the granularity of analysis and the perspective from which the analysis takes place, as an entity may be classified into a different category (or even multiple ones). Consider, for example, the Design Studio S drinks vending machine, available in selected locations in Japan[2]. On one level, the interaction with the vending machine is straightforward. A person (Actor) approaches the vending machine, chooses a drink (Object) to buy, uses the vending machine (Object) through a touch-based interface to purchase the drink and leaves.

However, this vending machine actually also incorporates intriguing functionality, which can change the perspective on the scenario completely. The machine does not show the drinks selection at first, forcing the person to come closer to the vending machine. Once the person is close enough, the machine captures an image of the person with the purpose of analysing it for demographic and sales data. Once the image is captured and analysed, the machine shows a drinks selection based on the results of the image analysis. The person is allowed to proceed with the purchase of a drink and their purchase data is linked to their image data. Once the person completes their purchase and leaves, the machine stops showing the drinks selection again, until the next person approaches the machine.[3].

Approaching the interaction scenario from this perspective, both the machine and the person show the characteristics of both an Object and an Actor. The machine is an Actor in that it actively controls the interaction by forcing the person to perform actions that allow the machine to achieve its own interactional goals. At the same time, once the machine's interactional goals are reached, it behaves more like an Object being passively manipulated through a touch interface. The person is an Object in that they are forced to perform certain actions without which they will be unable to reach their interactional goal. However, they also show Actor characteristics in that even while being forced to perform certain actions, they do retain the ability to disengage from the interaction as long as they are willing to forgo their goal of purchasing a drink. Once the machine's interactional goals are fulfilled, the person becomes a full Actor in the scenario.

The above scenario conveys the importance of selecting an appropriate perspective and level of granularity for analysing scenarios. However, within the context of this thesis the situation is generally more straightforward. System input and output devices (displays, interactive surfaces, keyboards, mice, etc.) are generally classified as Objects, people are almost exclusively Actors and Environments tend to correspond to rooms or workspaces. Where an entity exhibits characteristics of multiple entity classes, the extent of control over the interaction is the primary classification metric (e.g. if the entity exhibits characteristics of both Actor and Object, it is classed as an Actor).

[2] `http://www.fastcodesign.com/1662222/high-tech-vending-machine-is-a-full-on-robo-salesperson-video`

[3] Examples of other *dark patterns* of proxemic interaction can be found in [GB+14] and [BG+14] (cited in the Introduction as [C.5] and [J.2])

2.2 Relationships

By considering the spatial relationships between entities, the IRE model enables the analysis of a variety of spatial interactions within a system. Additionally, it offers the opportunity to decompose how each of the spatial relationships is utilised. The relationships between the different entities are considered from three dimensions *Position*, *Distance*, and *Orientation*, with each of the dimensions capturing a distinct perspective on the relationships.

2.2.1 Position

Position is intuitively a very straightforward measure. However, its use within the IRE model is somewhat subtle. The majority of the systems that use spatial information for interactive purposes use a tracking method that reports a 2D (x, y) or 3D (x, y, z) position of tracked markers/people/objects. By extension it could be said that all those systems use position. However, this is not true, at least within the constraints of the model, as most of the time only a subset of the positional information is used (most commonly distance). The use of the positional information for determining distance between entities falls under the distance dimension. Deriving orientation from positional information falls under orientation. The position dimension covers cases where the positional information is used to determine a more complex relationship, which cannot be described by distance only or orientation only. For example:

- The position of an entity A along a line running perpendicular to entity B's front (a person moving left to right along a wall sized display, but keeping constant distance from the display)
- Whether the positions of entities A, B and C form a line (one person occluding another person's view of a display by standing in from of them)
- Position of entity A relative to entity B's display surface (for an interface where a small high-resolution mobile display provides a detailed view of information shown on a larger but lower resolution display)

While most systems internally use a coordinate based description of position, there are several ways of describing position linguistically in addition to expressing it using coordinates. In the English language, three frames of reference are used. The descriptions of the reference frames will use the following terminology. A primary entity is the entity whose position is being determined. A secondary entity is an entity used as a primary spatial reference. The observer is the person verbally describing the position.

The *intrinsic* frame of reference uses the primary and the secondary entities to describe the position. It uses the intrinsic properties of the secondary entity to describe the spatial relationship. For example "the person is in front of the house" or "the sofa is in front of the TV", where both the house and the TV have defined front sides. In some cases, the primary entity's intrinsic properties may be used instead of those of the secondary entity (e.g. "the TV's back is to the wall'.

The *relative* frame of reference requires knowledge of three points - a primary entity, a secondary entity and the observer. The position is described from the perspective of the observer, using relative terms, for example "the pen is to the left of the glass" or "the person is in front of the pillar"). Note that the secondary entity is only used as a spatial reference, without exploiting any of its specific characteristics. The *relative* frame of reference is most commonly used when either one or both of the entities do not have easily defined or understood sides or other characteristics enabling spatial reference in the dimensions used to describe the spatial relationship (e.g. it is difficult to describe the "front of a pint glass").

In the *absolute* frame of reference, the relationship between the primary and secondary entities is described with respect to a specific fixed point in space. A typical example is describing the relationship using cardinal directions (e.g. "the person is north of the house") or the intrinsic properties of the fixed point (e.g. "the village is downstream from here", "the cottage is downhill from here"). Like the *relative* frame of reference, no intrinsic characteristics of any of the entities need to be defined or present as an independent fixed point provides spatial reference. Moreover, similarly to the *intrinsic* frame of reference, the spatial relationship can be described independently of the current location of the person describing the relationship.

Position relationships within the IRE model are not directional because while the linguistic descriptions are likely to differ if the position is described using entity A as the primary entity and then entity B as the primary entity, the actual positions still remain the same and the same result in terms of positioning can be derived from either description. When considering positions described using a coordinate system, this becomes much more obvious. Therefore, there are only six possible positional relationships: Actor-Actor, Actor-Object, Actor-Environment, Object-Object, Object-Environment, and Environment-Environment.

2.2.2 Distance

Distance is a measure of the relative (Euclidian) distance between two reference points. The points may correspond to any of the entities relevant to distance based relationships. Since the distance will be identical in either direction, the relationships captured in this dimension are not directional.

The first consideration when describing distance relationships is the choice of reference points. In most cases, distance is measured either from the centre of an entity or from a point on its surface or boundary. For the surface or boundary distance, the chosen point is generally the point closest to the other entity or a point where the surface/boundary intersects an imaginary line between the centres of the two entities. Additionally, for an Object, distance is sometimes measured from the point on its surface most relevant to the interaction (e.g. for a distance from a touch surface, distance from the centre of a display surface). For an Actor, distance is generally measured from the most relevant point for a specific modality (e.g. the eyes/head for visual output of a system or fingers/hands for finger/hand based inputs such as touch inputs or mid-air gestures). In cases of non-human Actors the principle applies in an analogous fashion. For an Environment, the most commonly measured distance is the distance from its boundary (to determine presence/absence of another entity).

In human descriptions, distance tends to be described in the *intrinsic* frame of reference, requiring only entities A and B. Since it is the distance between the entities that is the desired information, no specific information about the entities is generally necessary (e.g. "the person is standing two metres from the display"). However, with larger or more complex entities their intrinsic properties are sometimes used in the description (e.g. "the person is standing one metre from the boot of the car").

The possible distance relationships are Actor-Actor, Actor-Object, Actor-Environment, Object-Object, Object-Environment, and Environment-Environment.

2.2.3 Orientation

Orientation is a measure of the relative angle to an entity from the reference direction of a second entity. The measure is most commonly an angle within a single specific plane, but in some cases orientation within three dimensions (roll, pitch, yaw) is used. Unlike distance, there is no requirement for a reciprocal relationship between the entities and therefore the relationships are directional (e.g. entity A may be facing entity B but B may be facing away from A simultaneously). In this thesis, the primary entity, whose orientation we are trying to determine, will be named first within the relationship (e.g. Actor-Object orientation is the angle of a Actor towards/away from an Object).

Orientation can also be described without the use of angles by using a linguistic description instead. In English, the *intrinsic* frame of reference tends to be used when describing orientation. The general descriptive pattern uses two entities A, B, where A is the entity, whose orientation is being described, and entity B is used as a spatial reference point. While the intrinsic properties of either entity are not always used in the description, they can be used with more complex situations or to increase the expressiveness of the description. Expressions such as "the person is facing towards the display", "the person is looking to the left of the table" or "the display is facing the entrance to the room" are examples of both lower and higher complexity descriptions. Sometimes the *absolute* frame of reference may also be used (e.g. "the display is facing north-east").

The possible relationships for describing orientation within the IRE model are Object-Object, Object-Actor/Actor-Object, Actor-Actor, Object-Environment/Environment-Object, Actor-Environment /Environment-Actor, and Environment-Environment. All the relationships between Objects, Actors, and Environments are generally straightforward relative orientations. Note that when orientation is described in absolute orientation (roll, pitch, yaw), it is not necessary to use another entity to describe it. The same effect can be achieved using cardinal and relative directions within the *absolute* reference frame, for example "the person is facing northward" or "the person is looking downward".

2.3 Interaction

Arguably the most complex part of the IRE model deals with interaction characteristics. Five principal measures of interaction are considered: *range*, *mode*, *intentionality*, *intensity*, and *cardinality*.

2.3.1 Range

As the name suggests, the main concern for this perspective is the range of interactions. Range refers to the range of distances at which interaction is possible. In the thesis, this measure is used to capture the theoretically possible interaction ranges for a given system. However, the measure can also be used to describe the distribution of ranges in use over time to establish the ranges at which the system is most commonly used.

The ranges considered for this measure mirror the four proxemic zones as defined by Hall in his seminal work on proxemics [Hal66] - *intimate*, *personal*, *social*, and *public*. However, while the ranges mirror the well known proxemic zones, they are not identical. The naming is retained to foster familiarity with the terms but the values used to define the ranges do not use interpersonal distance as their basis. Instead, the most frequently used interaction distances for common interaction devices and settings are used to define the ranges. Definitions of the four ranges follow:

intimate (<0.6 m) The intimate range is where we interact with our most personal devices and where the most intimate of human-human interactions take place. The distance range for intimate interactions is based on the most common interaction distances for mobile phones, which is approximately up to 60 cm [LR+14].

personal (<1.5 m) The personal range encompasses the primarily single-user personal systems (tablets, laptops, office desktop systems, touch-enabled tables). Additionally, interactions fall within the personal range when the interactive parts of a system are within easy reaching distance, defined as within arm's reach after no more than a single step, which corresponds to approximately 1.5 metres.

social (<5 m) The social range contains interactions, which commonly involve multiple people interacting with a system and tend to involve some level of sharing of resources (both in terms of shared space and the interactive system). While interactions on the social range do not tend

to be primarily through haptic modalities (e.g. touch), interactions in this range should not require augmentation or extensive effort (e.g. there should be no need to raise one's voice to be heard and one should be comfortably able to see the majority of the interactive space without significant changes of visual focus). The outer boundary for the social range is approximately 5 metres, corresponding to typical dimensions of a room, and close to typical depth of field when focussing at a distance of 2.5 metres, which is the mean viewing distance for TV sets [FK+07] (DOF approx. 1.5 m–4+ m).

public (>5 m) Interactions in the public range are defined as interactions involving distances greater than 5 metres. Interactions in this range generally require significant effort on the part of the human unless they are augmented with additional sensors or interact through Objects.

To summarise, the four presented ranges, analogous to the proxemic zones introduced by Hall [Hal66], approximately correspond to typical interaction distances with common interactive devices.

2.3.2 Mode

There is a great wealth of research in the area of multimodal interaction. The IRE model does not capture modalities at great detail partly due to the limited scope of this thesis but more importantly because the sensing methods are likely to change with time. Moreover, most interactions actually use multiple modalities even when they are traditionally classed within a specific modality class. Take, for example, keyboard input on a physical keyboard. A single key press involves the following actions and modalities (modalities highlighted in italics):

- positioning a finger on the right key using *mechanical movement* of the finger/arm/body, using *proprioception*, *vision*, and *touch* (for the final contact) as feedback modalities
- increasing the finger's *pressure* on the key (with *mechanical movement* to maintain the pressure level on the button once it starts moving) until the key reaches the switching threshold or the end of its movement range, usually accompanied by audible clicking sound (*audition*) and a change in the key's resistance to *pressure*.
- reversing the *mechanical movement* to release the pressure on the key until the finger no longer *touches* the key

As can be seen from the above example, unless the interactions are captured at a level of granularity that would allow essentially atomic interactions (single movement of a finger, single unit of visual output on the computer display), most interactions are multimodal. Additionally, sensing methods can change. For example, instead of a keyboard with physical buttons, a person can use a software keyboard on a touch-enabled display or speech recognition.

Therefore, the IRE model uses *modes* rather than modalities to avoid confusion. A *mode* primarily captures what type of information is sensed and/or transmitted rather than which particular sense/sensor is used to capture the information. *Symbolic* modes, which include all linguistic information are an obvious example of a mode class, which may use differing combinations of senses/sensors to convey the same type of information (speech vs. keyboard input vs. hand-writing vs. gestures/sign-language). Proprioception, which is a sense of relative positioning of body parts and movement effort, is another good example. While it is usually classified as part of the *haptic* modalities due to how it is sensed within the human body, it actually communicates spatial information.

The main mode classes and some of their constituent modalities are described below, ordered by their expected frequency of use:

visual shape, colour

intent pointing, holding (selection), placing (selection by placing finger on location), gaze (holding, placing, see [Cla03] for more details)

spatial position, orientation, proprioception, acceleration, movement, distance

symbolic speech, keyboard input, hand writing, gestures, body posture

auditory any non-linguistic sounds

haptic pressure, vibration, texture, contact

thermal temperature

chemical taste, smell

It is important to keep in mind that modes do not always correspond to modalities. The specific emitters/sensors may be very different in terms of traditional modalities, e.g. body posture (symbolic class in the IRE model) can be sensed through a depth camera (visual) but also using proprioception (spatial class in the IRE model). However, the information transmitted will fall within a specific mode class (symbolic). This is an important distinction as the focus is not on the transmission method but on the content. It is possible to have multiple levels of emitters/sensors from different classes involved, e.g. using sign language (symbolic information encoded as spatial information by the body), sensed by a depth camera (vision) and printed on a Braille printer (symbolic information encoded in a haptic medium). Theoretically, one could use the symbolic information to describe a particular sensory experience, thus encoding a different class of information again. However, an in depth analysis of multi-layered transmissions is outside the scope of the model.

2.3.3 Intentionality

Within the IRE model, intentionality measures the level of intent in a particular interactive act. Intentionality is described as a directional measure of intent from the emitter of the action to the interpreter of the action. In an example situation of a person pointing towards a display, the person is the emitter and the display is the interpreter. External intent is defined as the emitter intending for an action to be perceived by an entity. Intentionality can only be accurately measured on the emitter side because the same action can be interpreted in widely different manner under different circumstances.

There are three primary intentionality types for an action - *explicit*, *unintentional* and *ambient*. The primary intentionality types centre on the extent to which the emitter intended for the action to occur. An *explicit* action is any action that is directed externally towards another entity, for example a person pointing at another person. An *unintentional* action is one which may present externally, but which does not represent any external intention by the emitter. An example of this is a person absent-mindedly tapping their foot. An *ambient* action is in fact the absence of an action. Any meaning of this type of (non)action is solely inferred by the interpreter.

In addition to the three primary intentionality types, there are three secondary ones. These mainly exist to address two issues. Firstly, it is possible for an *explicit* action to be directed towards a specific interpreter, but the same action is simultaneously also interpreted either by another interpreter, or interpreted in multiple ways by the same interpreter. At that point the action is considered to be both *explicit* and *implicit*, as the implicit meaning of the action is also utilised for interactive purposes. Note that an action does not need to be *explicit* to also be *implicit*. An *implicit* action is simply an action, which is used for interactive purposes by an interpreter that is not the primary target of the action. Therefore, even otherwise *unintentional* or even *ambient* actions can also be *implicit*. The importance

of this becomes clear when considering Objects as they exert no active control of the interaction but their state can be used for interactive purposes. As an example, consider a mobile phone, which uses its orientation to switch between portrait and landscape modes of its display [HP+00]. The movement of the device is controlled by an Actor but it is the state of the Object that determines the display mode.

When evaluating actions, there is an inherent amount of error in the interpretations. This error may be induced by sensor inaccuracies, system or technique design, or incomplete knowledge about the entities or interactions taking place. An example of a common sensor issue is when a tracked person temporarily exits the tracked area and then returns. Most tracking systems generally assume that this new entity is a distinct individual, thus losing any interactional history and possibly misinterpreting their actions. Instances of the Midas Touch problem are most common examples of a system or technique design flaw. Consider a gesture tracking system that interprets any hand movement as a gesture, regardless of whether it truly is a gesture or not. Gaze-based systems also commonly struggle with distinguishing intentional pauses in gaze movement from general wandering of the gaze. An example of incomplete knowledge is a simple keyboard. Every key actuation is interpreted as intentional input into the system. This results in a situation where any key actuation interpreted as intentional, even if it is the result of an object accidentally placed on the key. In all these cases, actions are more or less likely to be misinterpreted. More importantly for the intentionality metric, it means that false positives are likely to occur. Therefore, two additional intentionality values are introduced - *explicit+* and *implicit+*. These types should be used when it is determined that while the system or a technique is based on an *explicit* or *implicit* action, the system/technique does not seem robust to interpretation errors.

To summarise, there are six intentionality types in total:

explicit An *explicit* action is an action performed with external intent specifically towards the interpreter. For example, a person pointing towards a display to select an item.

explicit+ In cases with significant likelihood of an erroneous interpretation of an action as *explicit*, the *explicit+* should be used instead, indicating that the seemingly *explicit* action may in fact be *implicit*, *unintentional* or simply an error.

implicit An action with *implicit* intent is an action where the interpreter is not the target of the action or where the action is used for purposes other than the primary intention of the action. An example of the first type of *implicit* action is a scenario where a person walks closer towards a painting so that they can see its details (the action is targeted towards the painting) but the action is also interpreted by a position tracking system, which emits an acoustic signal when the person approaches within arm's length of the painting. An example of the second type is a situation, where a person issues a speech command to a system (primary, *explicit* action) but the voice of the emitter is also analysed for pitch and volume to estimate their emotional state (*implicit* action).

implicit+ Similarly to *explicit+*, the *implicit+* type should be used in situations where the system designer's intention is to exploit an *implicit* action, but the interpretation process is likely to yield false positives.

unintentional An *unintentional* action is one with no external intention. Such an action can still be perceived as an input by the interpreter but the action is not intentionally performed by the emitter or it is performed with no intention of the action being perceived by any interpreter. An example of *unintentional* action could be a person absent-mindedly playing with a pen in their hand.

ambient *Ambient* intent corresponds to no action, so any information transmitted or perceived is contextual in character.

Since there is so much potential for differing interpretations of actions by distinct interpreters or under differing circumstances, examining the intentionality of action used within an interactive system is likely to yield valuable insights into the system. Moreover, considering the potential for misinterpretation of the intentionality, analysis of action intentionality and the assumptions made about actions could point to possible design weaknesses or even potential failure points.

2.3.4 Intensity

Similary to intentionality, intensity is another directional measure. It measures the impact of an action on an interpreter. Generally speaking it represents the magnitude and optionality of reaction to an action by an interpreter. Similarly to intentionality, there is also a continuum of values for intensity but five major classes are identified here (ordered from least intense to most intense).

unnoticeable An *unnoticeable* action has no impact on the interpreter and does not lead to any reaction by the interpreter.

subtle *Subtle* actions have some impact on the interpreter but the impact is generally small and the actions can be ignored easily.

neutral A *neutral* action has an impact on the interpreter. An action of this type is likely to lead to a reaction by the interpreter but any reaction is still voluntary. This category represents the highest intensity for actions that will not interrupt the interpreter if they are otherwise engaged.

intrusive As the intensity of an action increases, so does its intrusiveness. An *intrusive* action places a strong demand on the interpreter but it is still possible for the interpreter to avoid being completely interrupted with sufficient focus. Some interruption is unavoidable, however.

disruptive A *disruptive* action forces the interpreter to react strongly. After a *disruptive* action, the interpreter is not able to continue with any other activity of their own due to the disruptive nature of the action.

Like intentionality, intensity is directional in nature. Unlike intentionality, intensity can only be accurately measured on the interpreter's side of the interaction. Again, the same action can be perceived as having differing intensity by different interpreters and the intensity as intended by the emitter may not match the intensity with which the interpreter perceives the action. Also similarly to intentionality, examining the relationship between the intended impact of an action and the actual impact of an action can lead to useful insight.

2.3.5 Cardinality

The cardinality measure is based on classifying quantities of entities through their parallelism and connectedness. Table 2.1 concisely shows the notation that will be used in the remainder of the thesis. The notation can be applied to both people and displays. The *parallel* dimension denotes the multiplicity of independent entities. The *connected* dimension captures the multiplicity of groups of entities. If at least one group exists within an entity class, any other individual entity within that class is also considered to be a group for the purpose of classification (i.e. if there is a group of Actors and another independent Actor, they are considered to be two groups of Actors, simply with one of the groups being of size one).

In more detail:

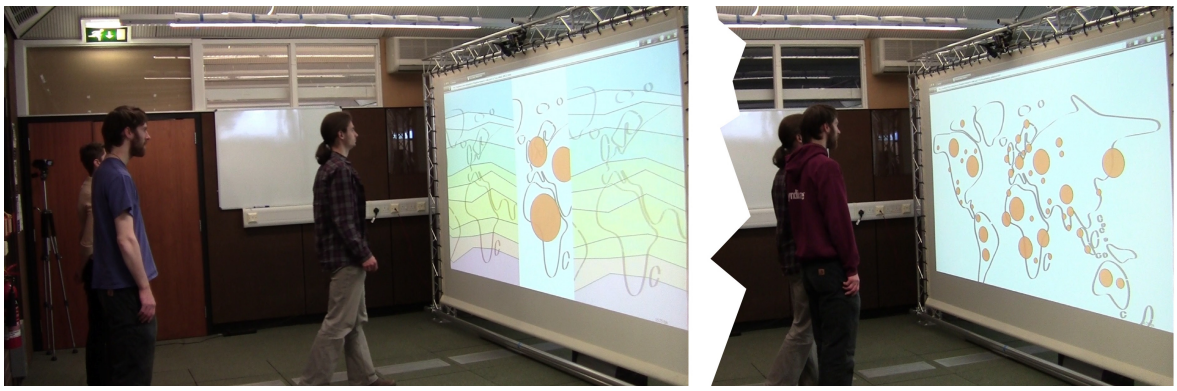


Figure 2.1: An illustration of the cardinality dimensions. The subfigure on the left shows three people interacting in parallel (`1*`). The subfigure on the right shows two people interacting as a group (`M`).

		Parallel	
		One	Many
Connected	One	1	1*
	Many	M	M*

Table 2.1: Overview of the multiplicity definitions.

`1` Starting with the most straightforward, `1` denotes a single entity.

`1*` A `*` denotes multiple independent entities of the same category, in parallel. Therefore, when applied to Actors, `1*` represents multiple Actors using the same system at the same time in parallel, but independently and without active collaboration. For Objects, this means multiple independent, uncoupled Objects present in the environment. For Environments, `1*` generally corresponds to a system being deployed to multiple sites, where each site is independent and has no connection to any of the other sites.

`M` A group of entities is classified as `M`. For example, a group of Actors collaborating on the same task would be denoted by `M`. A group can be an arbitrary size greater than 1. `M` is also used to represent multiple Objects that are coupled together as part of a task environment. For the purpose of this classification, the closeness of coupling between the entities does not matter, as long as the purpose of interaction is identical for all entities. For Actors, the classification tends to be straightforward with all the Actors collaborating on a specific task being part of a group, generally with one group per task. For Objects, the classification becomes more complex due to the coupling strength ambiguity. Within the IRE model, the coupling is generally decided on a task level as well, with all Objects required to perform a task by an Actor forming a group. For Environments, two (or more) Environments are considered a group when the Environments are connected in some fashion and one Environment can influence the other Environment, be it through interactions within the Environment or though their spatial relationship (distance/orientation/position).

`M*` As with `1*`, adding a `*` to `M` denotes parallel use. For Actors, `M*` signifies multiple groups of Actors with each group collaborating on a different task. In the Object context, `M*` represents an ecosystem, which consists of multiple sets of coupled Objects, with each set of Objects dedicated to a different task. For Environments, this class denotes multiple Environments, where some of the

Environments are interconnected (and thus forming groups) but not all of the Environments are interconnected, which means that at least two distinct Environment groups are formed.

The cases of mixed multiple sets are also collapsed into this category. This means that for the `M*` category, a group of entities is a group or a set greater or equal to one, rather than strictly greater than one for the `M` category. The motivation for this simplification is two-fold. Firstly, it reduces the complexity of the model. Secondly, for the very complex scenarios that form this category, the minimum number of group members matters far less than the number of groups and the overall number of entities involved in the interactions.

2.4 Discussion

The IRE model has been designed to offer a rich view of any analysed system and to provide many avenues for deepening understanding and gathering insights. This discussion covers some considerations for using the model for system analysis alone, or in conjunction with other interaction models.

2.4.1 Choice of Model as a Conceptual Construct

The distinction and specific definition of various conceptual constructs such as taxonomy, framework and model can differ between fields and in the specific case of HCI, which itself draws from many other fields (e.g. psychology, cognitive science, computer science), it can be non-trivial to discern the specific intended meaning.

To paraphrase Nachmias et al. [NN92], taxonomies are hierarchical structures of terms organised so that the relationships between terms can be expressed. Conceptual frameworks are theoretical constructs which place descriptive categories and the relationships between them in broader structure. Unlike a taxonomy which is primarily descriptive, a conceptual framework also provides explanations and predictions for empirical observations. Models are primarily representations of the structure and the features of phenomena. They are frequently also simplifications as well [NN92].

In the case of the interactive scenarios that are the focus of this thesis, I could have developed a set of taxonomies or an ontology (interpreted here as a conceptual construct allowing the expression of non-hierarchical relationships as well as hierarchical ones as in a taxonomy) instead of a model to accomplish the same goal. However, because there was certainly an element of simplification to reduce the complexity of the conceptual construct, it is more accurate to refer to the resulting construct as a model. The intended advantage of this simplification is a greater accessibility of the conceptual construct to practitioners and non-experts.

2.4.2 Analytical Considerations

Complex Relationships Where an interaction technique in an analysed system is so complex that is uses multiple relationships simultaneously, the technique should be classified under the spatial relationship that makes the largest contribution to the technique, with additional detail in the description of the classification. In cases where a complex interaction scenario is described, the scenario should be decomposed into constituent interaction techniques as much as possible.

Temporal, Input/Output and Other Analytical Dimensions The analysis in Chapter 3 uses a high-level analysis granularity in order to gather sufficient information from at times very limited descriptions of the analysed systems. However, while this may be necessary when only limited descriptions are available, the IRE model allows a much more detailed and granular levels of analysis to allow further insights into various systems. The model, as used in Chapter 3, examines the

maximum extent of values for each of its constituent parts, essentially performing a union of all individual values before determining analysis values. For example, if a system implements two techniques, one of which requires interaction in the intimate range and another at either personal or social range, the system is judged to use all three interaction ranges. This is even in cases where most often the second technique is only used in the social range.

If one is interested in a more granular view of a system and there is ample information available, it is possible to examine more subtle aspects of systems. Firstly, weighting all the values by their temporal distribution can reveal how frequently each of the system's features are used in practice. This can reveal various optimisation strategies as resources can be directed to parts of the system that are used most often. Moreover, it can also inform design decisions as techniques may be modified or replaced based on their burden on the system or the system's users, which could help increase the efficiency of the system as well as its usability.

If an even more granular view is desired. It is possible to analyse systems based on their inputs and outputs individually. Since the IRE model is designed as relatively independent of the notion of inputs and outputs, it is possible to decouple the two aspects of the system and analyse them individually. This approach can reveal interactive limitations and bottlenecks in cases where the intended inputs and outputs are not well matched (e.g. the content is optimally viewed from different distance than a person would be at in order to interact with it).

2.4.3 Other Models, Frameworks and Taxonomies

Over the years several models and taxonomies for classifying interactions have been proposed. One of the best known is the Groupware Time/Space Matrix (summarised in Table 2.2) introduced by Johansen et al. [Joh88; BJ88]. It was created to capture the types of interactions common in team based settings to help analyse their computational needs. It classifies group and collaborative interactions into four kinds based on their temporal and spatial characteristics. The main focus of the model is on team collaboration and the tasks that are to be completed, rather than the technological means to do so. The IRE model is complementary to the Groupware Matrix in that the focus of the IRE model is on the spatial relationships rather than the combination of temporal and spatial aspects. Moreover, it can be argued that the IRE model complements the Groupware Matrix by providing more in depth analysis of the spatial relationships far beyond co-located and remote collaboration.

	Same Time	Different Times
Same Place	Need: Face-to-Face Meetings Facilitation Services Computer-Supported Meetings GDSS	Need: Administrative, Filing & Filtering Presentation Aids Team Calendars Project Management Integrated Analysis Text Filtering
Different Places	Need: Cross-Distance Meetings HQ Conference Calls Graphics & Audio Screen Sharing Spontaneous Meetings	Need: Ongoing Coordination Group Writing Electronic Meetings Computer-Conferencing Conversational Structuring

Table 2.2: The Groupware Time/Space Matrix as it appeared in [BJ88]

In [TQD09], Terrenghi and colleagues introduce a multi-person-display (MPD) taxonomy for linking people with displays. Their taxonomy concentrates on two main dimensions - the size of the display ecosystem and the social nature of the interaction. Terrenghi et al. define the size of the display ecosystem as generally based on displays with maximum size one step down on the scale,

e.g. a *yard* scale ecosystem would consist mainly of *foot* size displays (see Table 2.3 for details). They acknowledge that the position and orientation of the displays affects their perceived size, so the specific ecosystem size is somewhat context dependent. The nature of social interactions is defined by a tuple consisting of the number of people driving the interaction and the number of people consuming the results (examples in Table 2.4). *One* is a single person, *few* are a group of three to nine people and *many* are a group larger than ten people.

Scale	Example	Display Size	Distance	Angle
Inch	Phone	3 cm	40 cm	4°
Foot	Tablet/laptop	35 cm	70 cm	28°
Yard	pub TV	1 m	3 m	19°
Yard	Tabletop	1 m	1 m	53°
Perch	Town centre	5 m	10 m	28°
Chain	Blinkenlights	20 m	50 m	23°

Table 2.3: The scale of single displays in relation to users' visual angle and distance as shown in [TQD09]

Scale	Example
One-one	Face-to-Face meeting between 2
One-few	Presentation in meeting room
Few-few	Around the table meeting
One-many	Leaving digital post-its on public displays?
Many-many	Sharing music in public

Table 2.4: The types of social interactions as presented in [TQD09]

The IRE model examines some of the same aspects of interaction as the MPD taxonomy, namely interaction distances and the magnitude of interacting entities. The *range* aspect of interaction in IRE model is related to the ecosystem size in the MPD taxonomy. However, where the MPD taxonomy focuses on the sizes of displays, the IRE model concentrates on the distances between the entities and the implications those distances have on the nature of the interactions. Similarly, the approaches to examining the magnitude of interacting entities differ in their focus. The MPD taxonomy provides a somewhat more granular description of magnitudes with regards to people (see Table 2.4) but other entities are not well covered. The IRE model, on the other hand, allows systems to be described on a more holistic level by allowing the magnitudes of multiple entity types to be captured. Moreover, the type of interaction can also be captured (collaborative interaction or independent parallel interaction).

The EasyLiving Geometric Model [BM+00] is a spatial model that centres around entities, similarly to the IRE model. However, unlike in the IRE model, entities are defined simply as objects in the physical world. Each entity has an associated set of measurements, which in the basic form includes the entity's position, orientation and an extent (in the form of a polygon). An extent can be used to describe physical attributes (e.g. size), or virtual properties (e.g. service area). Once the property measurements have been populated, the spatial relationships of the entity to other entities can be computationally queried, which in addition to positional information includes information about presence/absence within specific spatial regions or extent intersections. Whilst some of the terminology and concepts are shared between the EasyLiving Geometric Model and the IRE model, their focus differs. The EasyLiving Geometric Model is primary a computational model that allows spatial queries, while the IRE model is primarily an analytical tool.

Concepts similar to the *extent* in the EasyLiving Geometric Model are the basis of earlier work by Benford et al. The Focus/Nimbus Awareness Model ([Ben93; BB+93] and others). The model is

based around five core concepts - medium, aura, focus, nimbus, awareness and adapters. A medium is an interface that enables interaction between objects. This concept appears to be related to the concept of *mode* within the IRE model. Aura is defined as a region of space where an object may interact in a given medium. This is somewhat similar to the notion of *range* within the IRE model but specific to individual modes. Unlike in the IRE model, the aura is a region of space rather than a distance measure. Focus and nimbus are closely related concepts, where focus is a region of space where an entity A is aware of other entities, while nimbus is a region of space where other entities are aware of entity A. The various combinations of the focus and nimbus fields define the levels of awareness between entities. Lastly, adapters can be seen as modifiers of the focus and nimbus fields. The Focus/Nimbus model was later extended by Rodden [Rod96] to use a graph structure, which allows for uses of the model in non-spatial contexts and richer expressivity for collaborative interactions.

The Focus/Nimbus model is very complementary to the intentionality and intensity measures within the IRE model as it is more expressive for analysing awareness between entities. Additionally, one of the interesting aspects of both the EasyLiving Geometric Model and the Focus/Nimbus model that could benefit the IRE model is the use of space in terms of regions. In its current form, the IRE model does not include a straightforward way of considering arbitrary regions of space within the spatial relationships. This could be a valuable extension of the IRE model.

Activity Theory as applied to Human-Computer Interaction (see [Bød89; Kuu95; Nar95] and others for details) is also highly relevant to the IRE model. Firstly, Activity Theory uses concepts similar to Actor and Object in the IRE model. In Activity Theory, an activity is seen as a *subject* performing an action on an *object* to achieve an outcome. The *subject* and *object* concepts are mostly analogous to Actor and Object entities within the IRE model. However, Activity Theory also includes the concept of a *tool*, which facilitates the transformation of the *object* during an activity. Within the IRE model, a *tool* would be classified as an Object as well.

Secondly, processes are modelled as a three level hierarchy within Activity Theory - *operation*, *action*, and *activity*. An *operation* is an *action* that has become habitual and does not require significant conscious effort. An *action* requires conscious effort still represents a short term process. An *activity* generally represents the top of the hierarchical structure and consists of a series of *actions* used to achieve a goal. This hierarchy of actions is not captured within the IRE model in its present form, but it would likely prove highly beneficial in terms of defining and managing the level of granularity of analysis using the IRE model.

Other conceptual constructs that may of interest to the reader include the Context-Aware Architecture by Schilit [Sch95], or work by Dix et al. [DR+00], which includes a number of taxonomies related to space and location as well as a design framework for interactive mobile systems. More recently, there is also the Context-Aware Framework and Toolkit by Dey et al. [DAS01] and the Expected, Sensed and Desired Framework by Benford et al. [BS+05].

2.4.4 Applications of the Model

Although the primary design goal of the IRE model was analysis of existing systems to provide detailed view of the use of spatial relationships between interactive entities, the model could conceivably be used to achieve other goals. This section provides three examples of alternative use for the IRE model.

Firstly, the IRE model could be used to perform a gap analysis for a set of existing systems with the goal to establish which aspects of spatial interactions have not been explored yet. Chapter 3 provides an example of a similar analysis of existing systems.

Secondly, the model can form a part of the requirements gathering process for a new system. Once a gap has been found or when a new spatially-aware system is being designed, the IRE model

can be used to define a set of constraints or requirements for the new system. For example, the different sensors may be required to provide specific tracking speed or accuracy. Interactions within specific context may require a certain amount of interactive space to be available. Or, the interactive properties of the entities within the system's use case may allow for cheaper or less resource intensive design if only some interactive properties are required. The *MultiView* prototypes in Chapter 5 are an example of a system, which enables spatial interactions at least in part without active tracking and by leveraging commodity hardware.

Thirdly, the model could be used for finding existing systems for comparison. For example, if a researcher creates a novel spatially-interactive system, they could use the IRE model to analyse its properties. This could be combined with the requirements gathering in the previous example to draw inspiration from already implemented systems, or to learn from the difficulties faced by the designers of an existing system.

2.4.5 Extensions and Future Work

Model Extensions While the IRE model is complete for the purposes of this thesis, several refinements and extensions could be introduced in the future. Section 2.4.3 highlights a number of related models, frameworks and taxonomies that could be used in tandem with the IRE model to achieve richer analysis. In addition to those, there are other areas, where the IRE model only achieves partial coverage.

For example, the notion of *feedback* or *intelligibility* are only partially and indirectly addressed in the model in the form of the intentionality and intensity measures. Feedback and intelligibility capture how a system can communicate to its users about its state and behaviour. The Intelligibility and Accountability Framework introduced by Belloti and Edwards [BE01] or the work and techniques developed by Vermeulen [Ver14] are good starting points for creating an extension to the IRE model that captures this aspect of interactions.

Another possible extension could be the separation of the concept of robustness and accuracy of sensing from the Intentionality measure. An approach similar to the design framework by Benford et al. [BS+05] may be appropriate. In the framework, they distinguish between expected, sensed and desired movements and highlight how there may only be a partial overlap between those categories of movements in an interactive scenario with a particular system.

Model Validation The IRE model has been designed to be deterministic. However, since the model has only been used for analysis in this thesis and since the analysis was only performed by the author of the model, the deterministic nature of the model is not yet validated. Such validation could be performed by asking a number of people to perform an analysis of a selection of systems using the IRE model and comparing the results of the analysis to establish the level of agreement between them.

However, when designing such a validation study or comparing the results of an analysis using the IRE model, one should be careful about the goal of the analysis. The goal of the analysis will influence the level of granularity at which analysis should be performed. This was highlighted in Section 2.1.4 but it bears repeating. The granularity of analysis will almost certainly have an effect on the level of agreement between multiple analysts, so it is important to ensure all the analysts use the same level of granularity if the results of their analysis are to be compared.

2.5 Summary

The IRE model allows for detailed description and comparison of existing systems, for analysis of potential constraints within systems as well as opportunities for further refinement, and even for

providing an alternative view on the results of analysis using other interaction models. The model centres around the entities that form parts of interaction scenarios, their spatial relationships as well as a more general view of the interaction characteristics. The model is used in the next chapter for a detailed analysis of a number of state-of-the-art systems as well as the prototypes produced for this thesis.

Chapter 3

Existing Systems through the Lens of the IRE Model

The IRE model provides a framework and a set of lenses through which it is possible to examine existing and future systems. It also provides a way of conducting a gap analysis on the current state of the art systems and interfaces that use space for interaction. This chapter contains such an analysis performed with three goals in mind. The primary goal is to generate a deeper understanding of the spatial and interactional features that have been used in the past. The second is to map out opportunities for future research by examining the gaps in existing research. The third goal is to position the systems presented later in the thesis within the existing body of research.

To examine state of the art research through the lens of the IRE model, a thorough literature survey was performed. Figure 3.1 visually describes the process used for selecting systems for analysis. The selected systems are believed to be a representative, if not complete, set of systems that use spatial information for interactional purposes. As mentioned above, the analysis also includes four prototype systems that are covered in detail in later parts of the thesis.

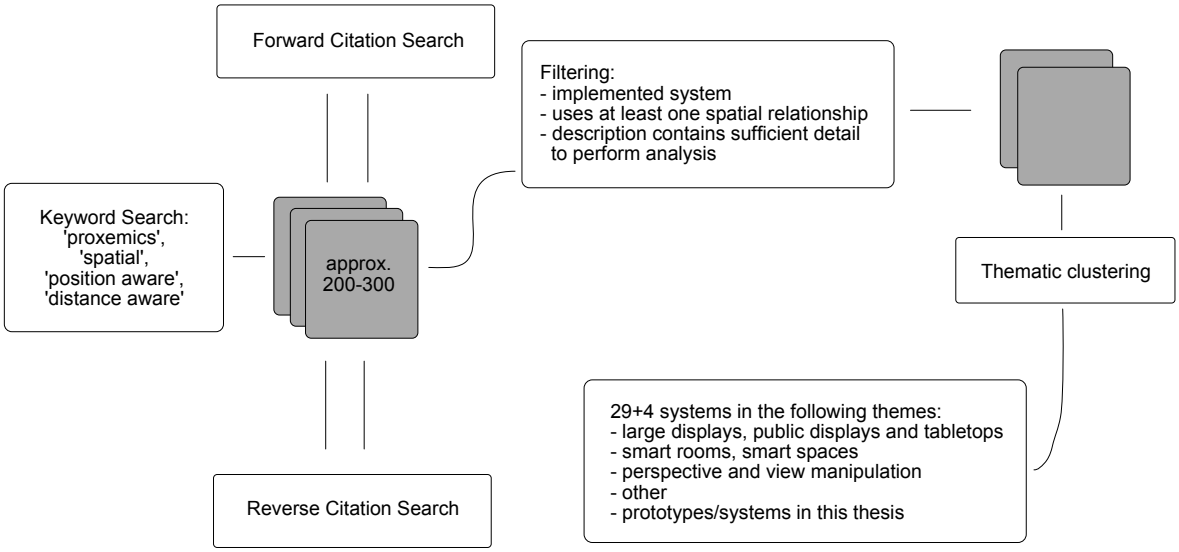


Figure 3.1: The process used to find and select systems for analysis in this chapter.

Each of the systems covered in this chapter will be referred to by a name. Where possible, the name used for the system is the one which the authors/developers of the system used in their descriptions and publications. Where such a name was not available, the system was named based on some of its primary characteristics. The name of each system is unique within this chapter and within the entire thesis (should the system be referred to in other parts of the thesis). References to publications used to gather details about the systems and to inform the analysis are provided where the system is introduced and in the relevant tables. Elsewhere, only the name of the system is used.

The chapter is structured as follows. First, all the systems included in the analysis are briefly introduced with focus on details that highlight characteristics relevant to the analysis. Following the introductions, the analysis proper starts by taking each of the three spatial dimensions in the IRE model in turn and scrutinising which entity relationships are utilised within the systems. A short discussion highlighting decisions taken during the analysis and addressing limitations related to the entity relationships follow. Then the focus shifts to the interaction components of the IRE model, again, using each of the components in turn as a lens through which the systems are examined. After another discussion on decisions and limitations relating to the interaction components comes a section that discusses the observations and trends that emerged from the analysis. This section also outlines potential opportunities for future research to explore some of the under-utilised relationships and characteristics.

3.1 Introduction to and Overview of Existing Research Systems

Before starting the analysis proper, all the selected systems will be briefly introduced. To make the introduction more structured, the systems are clustered around broader themes and topics. These clusters function primarily as a way to provide more structure to the introductions rather than as any strict or exclusive clustering.

3.1.1 Large Displays, Public Displays and Tabletops

The first set of systems examines various aspects of interaction with public displays, large and wall-sized displays and tabletops. Some of the systems concentrate on interaction techniques, some focus on sensing technologies, while others investigate higher level interactional concepts such as sharing and privacy.

Proxemic Peddler [WBG12; Wan12] is primarily a framework for designing attention-grabbing interfaces that use the distance and orientation of passers by to entice them to interact. The prototype system used to illustrate the principles in the framework is used for analysis. The system is based around a wall mounted display (52 inch diagonally, with touch capability) and a Vicon tracking system for tracking people. Since the system is actively trying to lure the person interacting with it into further interaction, the system is also considered an Actor within the IRE model, as well as the person being lured. The Peddler framework is based around trying to sell products in a manner similar to peddlers at local markets - using attention attracting techniques [WBG12]. The interactional flow includes techniques for re-capturing attention through movement within the person's peripheral vision. Additionally, the system uses gradual increase in detail about products to motivate the person to approach closer to the display and other similar techniques.

Is Anyone Looking? [BL+14] is a prototype system that is used to explore several techniques for managing over the shoulder peeking as a privacy concern when interacting with a large public display. The techniques that were implemented include flashing borders when a passer by is looking towards the display; indicating their gaze with a red dot and showing a body model of the passer by on the display; hiding content on a window level; and hiding content with (inverse) shadow casting. The system uses a Vicon tracking system to track people's position and head/torso orientation and

to position the display. This allows the system to calculate the distance and orientation between all the entities.

The *Puppeteer Display* [BB+14] is a wall-sized public display, which tries to manipulate people interacting with it into specific positions for interaction in multi-user scenarios (it tries to "actively shape the audience"). It does so using a variety of dynamic visual cues. The display also attempts to attract passers by. The paper includes two studies. The first evaluates visual techniques attracting passers by as they move parallel to the display (mostly concentrating on the positioning and size of the stimulus). The second study examines techniques for re-distributing users when multiple users use the display in parallel. The interaction scenario used in the second study was playing a game where users could bounce falling balls using their projected skeleton. On a technical level, the system uses a display (3.52×0.52 metres) mounted 1.25 metres above the ground level in a street-facing shop window. Two Kinect sensors were positioned below the display for tracking people.

With *Public Ambient Displays*, Vogel et. al. [VB04] introduced a framework for interacting with public ambient displays. The system in this work was used to demonstrate several techniques for interacting with public ambient displays. The techniques utilise and examine various aspects of interaction including public/private information display, attention-aware interaction, and touch and mid-air interaction. The underlying notion is that distance to display corresponds not only to different levels of engagement but also to different levels of privacy expectations in terms of information sharing. Interaction is zone based with four primary zones (based on distance) from furthest from display to closest: ambient display, implicit interaction, subtle interaction, personal interaction. Ambient display shows a range of glance-able information. The implicit interaction zone is activated when passers by are present. If a person faces the display (suggesting they are open to communication), the display shows an abstract representation of the person and a subtle notification of whether there is any urgent personal or public information waiting for them. The purpose of this zone is to attract the user and to initiate further interaction. The subtle interaction zone is activated when the passer by pauses for a moment, at which point more detailed versions of the notifications from implicit interaction zone are shown. Explicit interaction begins to take place in this zone (but distance-wise the interaction is still taking place more than an arm's length away from the display). The personal interaction zone is activated when a user selects an item/notification to interact with. Direct touch interaction is expected, and personal and private information is shown (as body occlusion provides some privacy). Technically, the system is based around a 50 inch wall mounted multitouch display. Vicon is used for tracking people (in volume of 8 feet in depth, 7 feet in height, and 16 feet wide).

Hello.Wall [SP+03; PR+03] can be considered an interactive art installation. The system consists of two parts - an interactive ambient wall display that shows a pattern of lights representing a particular set of information (tailored to the people that are in the vicinity of the display). The second part of the system if the ViewPort, which is a handheld device, which can show more detailed view of elements of the wall display. The primary information displayed within the system is either notifications and messages left by others or more general information about the site where the display is installed (e.g. company statistics and other information). *Hello.Wall* also uses zone based interactions. The ambient zone is outside of sensor range. The notification zone is an intermediate zone (where notifications can be shown on ViewPorts or on the wall display through light patterns) and the cell interaction zone is where direct exchange of information between the ViewPort and the wall display happens. The spatial sensing part of *Hello.Wall* consists of wireless radio sensors and RFID tags.

Psychic Space/Maze is one of the prototypes that are part of Myron Krueger's paper on responsive environments [Kru77; KGH85]. The Psychic Space uses a grid of pressure sensitive floor sensors. It is primarily a music instrument with each tile mapped onto different sounds, so the purpose of the system is to generate musical expressions. *Maze* is a repurposing of the environment and an application that enables a person to navigate a maze by moving on the floor. Closer to the display

is up, and so on. An interesting technique is used to counteract the fact that there are no physical boundaries in the real room. The computer changes the maze every time the person violates the constraints of the on-screen maze. Additionally, as the goal of the maze is approached, the maze is altered (the directions of physical movement are changed, so that front/back movement produces left/right movement on the display) or the symbol representing the user becomes stationary while the maze starts to move with the person's movements. In any case, solving the puzzle is not allowed (the maze is changed before the goal can be reached). On a technical level, the system is based around a room instrumented with a back-projected display (2.4×3 metres) and a grid of pressure sensitive tiles on the floor (4.9×7.3 metres).

The *Vision Kiosk* is the first of two prototype systems exploring interactions with public information kiosks [CA00]. The second system Agent Kiosk uses even less spatial information for interaction and is essentially a variant of the *Vision Kiosk*. Only the *Vision Kiosk* system is used for analysis as it serves well as an example and provides richer ground for analysis. The kiosk is based on a 21 inch display embedded in a kiosk body. Above the display is a camera used for tracking people. The system also contains a sound system for speech/sound output (speech control and recognition was only used in Agent Kiosk). The design of the *Vision Kiosk* was centred around an "avatar" (essentially an animated talking head), which provides useful information and entertaining comments.

The *Range Whiteboard* demonstrates the implicit interaction framework presented by Ju et. al. [JLK08] It is an interactive whiteboard designed to support co-located, ad-hoc meetings. It uses distance to transition between display and authoring modes, for clearing out space for writing, and for ink stroke clustering. Techniques examined within *Range Whiteboard* are user reflection (how a system can show what it senses/perceives/infers), system demonstration (how a system can show what it is doing), and override (how users can interrupt or stop the system from performing an action). The techniques are aimed at demonstrating increased interaction robustness and prevention of system induced interactional mistakes (erroneous proactive actions by the system). Technically, the system is based around a back-projected SMART board with four distance sensors mounted to the front of the board. Space in front of the whiteboard is divided into four zones. Intimate zone is used for writing on the board. Personal zone is defined as within arm's reach for users and expected use was pointing and text manipulation. Social zone was designated as easy viewing distance. Public zone was any space beyond the 1 m distance. The active zone at any given time was determined by the distance of the closest user detected as they are assumed to be driving the interaction.

Touch Projector is a prototype used to evaluate a remote interaction technique to manipulating content on distant displays [BB+10]. It uses live video captured by a mobile device's camera to identify and spatially register displays in relation to the mobile device. Any touch interaction on the display of the mobile device is then replicated on the corresponding remote displays. Additional variations of this technique to facilitate easier and more accurate transfer of content between displays are also evaluated. These include pausing the video stream to stabilise the target remote surface, magnification of the video stream to increase the accuracy of remote touch input and replacing the remote display contents in the video stream with locally rendered version of the content to increase the visual quality of the display contents (on the mobile phone).

Shadow Reaching [STB07] is an interaction technique for extending pointing and selection reach when interacting with large displays by applying perspective projection to the shadow of a person. The paper introducing the technique also contains three prototypes that used variations of the technique. The first prototype uses a real-world shadow projected on a display for use as a selection point (using a Polhemus position tracker and a button for sensing the actual position of the user's hand). The second prototype uses the entire body shadow for whole-body input in a game where the user interacts with bouncing balls using his/her shadow (the shadow is virtual — the shadow's shape was extracted from IR camera data). The third prototype uses virtual shadows as magic lenses to show different visual data (satellite map instead of vector map). The third prototype uses the

same method of sensing as the second prototype. While the first prototype is primarily used in the analysis, the properties of all three prototypes are nearly identical in the IRE model.

Medusa [AG+11] is a multi-touch table (based on the first version of the Microsoft Surface table) augmented with IR proximity sensors on the top and sides of the table. The proximity sensors are arranged in three rings: outward facing ring on the table's edge (long range) for detecting presence of people, outer ring on the top side (long range) for detecting arms, inner ring directly next to the multitouch surface for finer grade arm/hand/palm detection. While a large part of the paper introducing *Medusa* concentrates on the specifics of the implementation of the system, the paper also contains a sample application Proxi-Sketch that leverages the system's design. Proxi-Sketch is a prototyping environment for graphical user interfaces (similar to Balsamiq or Axure), but the focus is on multi-user interaction techniques such as hover interaction, user identification and widget ownership, and functionality being spatially constrained to specific sides of the table.

ProximityTable [Aln15; HR+14] is another spatially-aware tabletop system. It uses a ceiling mounted Kinect to track people around a 70 inch tabletop display. Only two sides of the table are tracked. The tracked area extends to 60 cm from the table edge for one side and 40 cm from the edge for the second side. Again, a significant portion of the description was based around the implementation of the system and less focus was placed on the interactions. The sample application used to evaluate the system was an application that allowed users to browse through a museum catalogue.

This concludes the introduction of systems in the theme focusing on large displays, public displays and tabletops. The smart space/smart room cluster is described next.

3.1.2 Smart Rooms and Smart Spaces

The second broad theme brings together research and systems that examine some aspect of interaction on a room-level or even larger scale. There are several large infrastructure-based projects and a number of systems exploring the notion of a smart room. While most of the systems in this theme focus primarily on people, some systems explore mainly device-to-device interactions instead.

One of the device-focused systems is *GroupTogether* [MHG12], which is a system that uses f-formations (group shapes formed by people engaged in face-to-face interactions) and micro-mobility (tilting of devices towards each other by people collaborating so that all involved can see what the actor is trying to show) to explore cross-device interaction techniques. The system consists of tablets with multitouch screens, augmented with phidgets motion sensors and radio modules. A 50 inch SmartBoard was mounted on the wall and used as a shared display. Two ceiling-mounted Kinects are used to track people and radio modules are used to track devices. In this system, an environment corresponds to an f-formation rather than the room. An f-formation is assumed when two people are not standing behind one another, they do not face away from each other, their distance is "small enough to comfortably communicate", and their o-spaces overlap. Types of f-formations tracked are: L-shaped, face-to-face, side-by-side, none. The primary focus of the research is techniques that allow people to share content on their tablets during co-located collaboration.

Proxemic Controls [Led14; LGB15; GL13] also focuses on devices and objects but in this case, the interest lies in control of appliances and devices using the notion of proxemics. The centre piece of the system is a mobile device, which acts as a universal controller. Depending on distance, position and orientation of the controller, different appliances/devices can be interacted with in various ways. Possible interactions use the gradual engagement pattern, where the distance from the object defines possible actions. Taking a thermostat as an example, from afar the controller shows the room temperature, coming closer, the controller shows current setting of the thermostat (still read only), coming closer yet the user can change the temperature settings and when very close to the thermostat the daily schedule is revealed. On a technical level, the system uses the Vicon tracking

system for tracking the position of the tablet. The appliances were custom made with an ability to be controlled over the network (lamp, radio, television); the remaining appliances were digital simulations.

Proxemic Media Player [BMG10; GM+11; Mar13] is an example application used to demonstrate how Hall's notion of proxemics [Hal66] could be leveraged by interfaces within smart rooms. The conceptual proxemics framework defines five dimensions of proxemics - position, orientation, movement, identity and location. The application scenario used to demonstrate examples of the concepts is very rich and includes a broad variety of techniques that touch on using distance to a display, orientation to a display, orientation of people to other people, dynamically repurposing devices and objects based on their spatial relationships with people. The research even deals with some cases of simultaneous interaction by multiple people. Due to the number of techniques, they will be discussed in more detail in the relevant parts of the analysis. The smart environment consists of a living room style room instrumented with a Vicon tracking system. There is a shared large display mounter on the wall, a mobile phone, a tablet, a spatially tracked pen (as an example of a non-digital object), and one to two people.

Code Space [BD11] is a smart room system, which concentrates on supporting co-located small-group developer meetings. It does so by enabling cross-device interactions and touch/in-air interaction. The system is based around interconnected laptops, phones, tablets and a large shared display. This is coupled with a Kinect for people tracking. Most interaction happens through remote pointing, gestures or touch. However, some more complex interaction techniques are employed as well, e.g. holding up a phone vertically towards the shared display to show the contents of the phone temporarily on the shared display. The smart environment consists of a 42 inch shared display with two-touch IR input. Kinect sensors are used for skeleton tracking. One mobile phone and one tablet are provided as mobile devices. Matching users to devices is based on who is logged on to each device. The application scenario revolves around software developers performing a code review.

LightSpace [WB10] takes a very different approach to creating a smart space. It combines multiple Kinects and multiple projectors to create an interactive space that allows touch and mid-air interaction. A notable characteristic of the system is that any surface (including people) can be a display and any surface can be touch enabled. Connections between different surfaces initiate transfer of selected objects. Digital objects can be scooped up into one's hand as a ball, which exhibits similar behaviour to a real ball (rolling along sloped surfaces and so on). The digital object can be transferred in ball form (even between people). Another notable technique is the spatial menu, which allows users to browse and select items from a vertical menu by moving their hand up/down above the menu point. Selection is made by remaining in a static position for an interval of time.

EasyLiving [BM+00; KH+00; BS09] is a project whose aim is to create an intelligent environment with a large number of interconnected and interactive devices in such a way that the devices will produce a coherent user experience. It is a large project with several publications examining different aspects of the project. The project facilitates device identification (and selection) through a spatial model (EasyLiving Geometric Model). It also includes a service discovery framework (InConcert middleware), which allows for dynamic assignment and/or configuration of devices based on their capabilities and the needs of the interaction scenario (though the papers generally describe it as an addressing and message passing middleware). An interesting aspect of the system is that it allows peripherals (keyboards, etc.) to be dynamically linked to other devices rather than being statically linked to a specific device. The downside of this approach, as mentioned above, is that identifying relationships between devices (and people) becomes more complex. The geometric model concentrates on in-home and in-office tasks with multiple I/O devices and possibly multiple users. Most of the tracking uses computer vision, either using stereo cameras for people tracking or using simple RGB cameras for device tracking. Object tracking is also possible using radio signal strength triangulation. Identity of people is established either by using a fingerprint reader or by logging

into a device. Four sample applications were developed to demonstrate some of the capabilities of the resulting system. Room Controller provides an overview of all the available services to a person and allows them to take actions (e.g. turning lights on). Remote Sessions is an application that automatically (based on spatial relationships) or explicitly (on a person's direction) moves the person's desktop to a display. Mouse Anywhere automatically directs the mouse input of a radio frequency mouse available in the environment to a person's display if there is only one person in the environment. The Media Control application automatically loads custom media (music, specifically) preferences into the system when a person is authenticated into the system. Details of the *EasyLiving* project are also discussed in other publications [SKB98; KSW01; HBB02] but their content does not add any significant fact that would be useful in this analysis.

Moving onto large scale, infrastructure-based systems, Want et al. [WH92; WH+92] introduce two systems in their papers. The Active Badge and a more advanced version called the *Authenticated Badge*. In both cases, the badge has a transmitter, which signals a unique code to a network of sensors distributed throughout the environment. A location server can then establish where the badge is located. The *Authenticated Badge* extends the Active Badge by including a signal receiver so it can be challenged to produce a unique response. This is used to verify that the badge is a genuine one. The environment can be equipped with radio emitters with a range of several feet, which will automatically trigger the badge. If a button on the badge is pressed within this field, the security challenge/response protocol will be initiated. The *Authenticated Badge* system is used in this analysis as it is the more complex system of the system and offers a richer set of possible interactions. Applications for an *Authenticated Badge* include a secure door entry systems (radio field automatically triggers challenge/ response), workstation login (same mechanism). Or, if a phone call is routed to a phone, an LED on the badge of the intended recipient lights up to help identify who the call is for. Applications that are shared with the Active Badge system include an overview screen at the receptionist desk, which shows the last known locations of all tracked people and the closest telephone extension to their location so that telephone calls can be routed to where they currently are rather than to their office. The *Authenticated Badge* has a radio detector (for detecting short range fields), IR emitter/receiver (challenge/response), LEDs, two buttons, and a beeper. The IR receiver/transmitter has a range of approx. 6/10m (the two papers differ on the range) and IR reflects off objects, so no directional information available.

Context-Aware Applications [SAW94] is an amalgamation of applications using the ParcTab as their basis. The ParcTab is a mobile handheld device with a 128×64 pixel display, a small speaker and an IR emitter/receiver. There are also three buttons for input and the screen is touch sensitive. The ParcTab emits a tracking beacon packet regularly, which means it's location within the building can be tracked. Related stationary devices can also emit beacons to indicate their presence/locations. Four applications areas are described. The first application area is Proximal Selection. It is used to select the closest input/output device to the ParcTab. This includes printers, displays, speakers, video cameras, thermostats and so on. This could also be people, used as targets for data transmissions. It could also be non-physical objects accessed only from specific locations such as bank-statements, menus, manuals and so on. The second application area is Automatic Contextual Reconfiguration. This technique can be used to trigger a binding between the tabs and a virtual whiteboard which can show notes that were left in that particular room. Moving to another room would show that room's whiteboard instead. The third application area is Contextual Information and Commands. It works on the assumption that some of people's actions can be predicted from context (e.g. different tasks are performed in the kitchen and in the library). The location browser application on the tab shows a location-based filesystem, which switches between folders based on which room people are in. The fourth application area is Context-Triggered Actions. These are if-then rules that get triggered based on a set of conditions. The difference is that these actions are triggered automatically without user input (at the time of the action). There were two example applications. One was based on the Active

Badge (mentioned above) and the other on the ParcTab. The badge based application was called Watchdog and it triggered arbitrary unix commands when a certain location was reached. The tab application was called Contextual Reminders and it showed a visual reminder on the tab when a set of conditions was fulfilled.

The *Sentient Computing System* [AC+01; HH+02] was a prototype context aware system, where objects within the environment react to people's presence and their actions. The system was installed in a three floor, 10,000 ft^2 office building with 50 continuous users. The sensor network uses 750 sensors and three radio cells and tracks 200 Bat units. Location sensing is performed using Bats, which are sensor units that can be carried around or attached to equipment. A Bat is a $8 \times 4.1 \times 1.8$ cm device with two input buttons, a buzzer, and two LEDs. It also contains an ultrasonic emitter and a radio receiver. Radio signals will cause the Bat to emit ultrasonic pulses, communicating its location with accuracy to within 3 cm. The Bat also contains a motion detector. A person's orientation can be sensed because the wearer's body occludes part of the emitted signals and therefore it is possible to infer coarse grained orientation. It is possible to detect whether the wearer is standing or sitting as the distance from the ceiling sensors can be computed. Three application areas were described in the paper - Follow-me Systems, Novel User Interfaces, and Data creation/storage/retrieval. There is also an overview application called Browsing, which displays a 2D map of the office building with spatial representation of all tracked people, objects (workstations, telephones, but also desks) and spaces (rooms, corridors, etc.). The map can be queried for information about finding people, locating nearest phone, or for setting spatial reminders. Since the number of applications is rather large, only a subset will be mentioned here and more details will be provided where appropriate during analysis. An example from the Follow-me Systems area: when a person's telephone rings and they are not in their office, the Bat makes a sound. By pressing a button, the person can forward the call to the nearest telephone. An example from Novel User Interfaces: positioning the bat in the "mouse panel space" projects the bat's position onto a 50 inch display nearby, so it can be used as a mouse pointer. And lastly an example from the Data creation/storage/retrieval area: since bat augmented objects can be located as well, it is possible to infer who holds them. Such devices can be personalised on being picked up, data storage can be routed to specific storage systems (store user's files on their drive), or additional information can be added to the device based on context.

This concludes the introduction of systems in the theme focusing on smart rooms and smart spaces. The perspective and view manipulation cluster is described next.

3.1.3 Perspective and View Manipulation

This section introduces systems, whose primary functionality focuses on manipulating the view of displayed content, whether it is to correct perspective, provide more detail or to offer alternative views.

Starting with the simplest of the systems, *Lean & Zoom* [HD08] is a prototype application running on a laptop with a 13 inch display, which uses distance of a person in front of its display to magnify on-screen content. The person is augmented with white markers. These markers are detected by the computer's camera and based on the distance between the two markers the distance of the person from the display is established. While the primary technique is to magnify content on the display, an alternative technique that uses semantic zoom is also demonstrated.

Moving to a more complex system, *E-conic* [NS+07] is a perspective-aware multi display environment. Within this environment, all windows and other UI elements can be perspective corrected, so that they are always easily readable for the person the UI elements belong to. Beyond perspective correction, the apparent size of windows can also be kept constant. Windows can also be rendered on multiple displays simultaneously. The system consists of a number of displays, each running a client application, connected to a geometry server. Content is received from the application server,

and each client is responsible for rendering content at positions/sizes/shapes appropriate to fulfil all positional constraints. Clients are also responsible for transmitting user input (using perspective cursor). The geometry server has all static element positions encoded a-priori. Mobile displays (laptops) and people's heads are tracked using an ultrasonic tracking system (with inertial tracking included for mobile displays as well to increase tracking quality). Configurations tested included up to 5 displays (4 static displays and a mobile tablet), all connected using wired connections. However, a study evaluating the system used a 3 display configuration with a bottom projected tabletop display, a large vertical display and a standard monitor positioned on top of the tabletop.

Screenfinity [SMB13] is a system that validates a model of perception area for content on large public displays. The aim of the model is to maximise the area, from which information can be read by users and passers by. The model is based around three content modification techniques — zoom/magnification, horizontal translation and rotation (perspective correction). In the preliminary laboratory studies, a 5×2.5 m back-projected display and the OptiTrack tracking system were used. For the field study the same display was used but tracking of people was performed by clusters of Asus Xtion Pro depth cameras (Kinect equivalent). While the techniques are the same for both versions of the system, where the systems differ the field study version is the one primarily used for analysis.

With *Proxemic InfoVis*, Jakobsen et al. [JS+13; JH12] use the proxemics view of interaction to map out the design space of proxemics in information visualisation and to propose a number of techniques. They also implement an example system to test out some of the proposed techniques. Analysis is based on the implemented techniques (separate user studies aggregated into one system for purposes of analysis). The research includes three studies. Study 1 examines different possible zooming and panning techniques (absolute movement, relative movement). Study 2 examines a distance-based aggregation technique, which varies details shown based on distance to display. Study 3 investigates use of position along the display for selecting a subset of the shown data (movement mapped to the x-axis of the visualisation), while also using distance to select attributes on which the selection is performed. Technically, the system is based around a 3×1.3 m back-projected display and an OptiTrack tracking system for tracking people (specifically people's heads).

Egocentric ZUI - Rädle et. al. [RJ+13] performed two experiments examining the use of egocentric navigation with zoomable user interfaces on large displays. The analysis will concentrate on the prototype used in the egocentric navigation condition. The experimental setup included a large wall display and a tablet device. The tablet was tracked with the OptiTrack tracking system. Both the display and the tablet showed parts of the same canvas. A purple rectangle was also displayed on the wall display to represent the view currently visible on the tablet screen. Panning and zooming was performed using spatial movement in relation to the wall display (front-back movement controlled zoom, while left-right-up-down movement controlled panning). Zooming was setup such that 0.75× magnification factor was reached at 190 cm from the wall display and 8× magnification factor was reached at 30 cm from wall display. Memory card content was only shown with magnification factor greater than four and it was only displayed on the tablet, the wall side display only showed the back of the card. Content was distributed in such a way that at most one memory card would be visible at a time on the tablet. The experiments showed a noticeable decrease in path length (47%) and task completion time (34%) compared to the baseline multitouch condition.

This concludes the description of systems in the theme focusing on perspective and view manipulation. The systems that did not fit into any of the previously described clusters are described next.

3.1.4 Other

The systems presented in this section do not form a coherent theme but they still use spatial information for interaction purposes. Two of the systems concentrate on interpersonal communication, while the third system focuses on interactions between a person and a mobile device.

Opo [HK+14] is a wearable sensor system for tracking distances between people, with a particular focus on tracking face-to-face interactions. Most of the paper describing the system concentrates on the design of the sensors used within the system. The primary motivation was to enable tracking of interactions in large (or multiple) environments without any installed infrastructure. Because the sensor is assumed to be unique to each individual and non-transferable, the resulting system is essentially a people tracking system. The sensors use a combination of radio and ultrasonic emitters and receivers to estimate distance between the sensor units within its field of view. This allows the system to track, who the wearer of the sensor unit is interacting with, and at which distances.

Active Hydra [Gre99; KG99] is a prototype system presented as part of an exploration on digital/physical surrogates for communication and remote collaboration. The *Active Hydra* consists of a screen, speaker, video camera, microphone and a proximity sensor. The concept is based on Bill Buxton's' Hydra surrogates[1] but it extended with notions of proximity. Distance of the remote collaborators to the Hydra units determines the quality of the audio/video link. The system can be augmented with other surrogates (peek-a-boo surrogate, responding surrogate) to explicitly communicate availability or expectations of privacy. In the paper, the Active Hydra system was mostly used in combination with a peek-a-boo surrogate and with a responding surrogate, so the combination of the three surrogates is used in this analysis (it also makes the over system richer in terms of possible interactions and the use of spatial information). The orientation of the peek-a-boo surrogate (a figurine) communicates presence of a remote person: facing the wall means the person is not present, facing forward means the person is present, in between orientation states communicate how much time has passed since last activity. The responding surrogate denotes a person's willingness to communicate: a figurine on stage means the person is very interested in communication, away from stage means they are open to communication, and no interest is indicated by tipping figurine over.

Mobile Sensing Techniques - In 2000, Hinckley et al. [HP+00] introduced an augmented handheld device for experimenting with interaction techniques using orientation and proximity of the device and its user. The PalmPC was augmented with an IR emitter/receiver for sensing distance, a tilt sensor for sensing device orientation and a touch sensor along the side and back of the device. The tilt sensor can detect whether the device is laying flat or whether it is being held vertically. The proximity IR sensor is used to detect essentially three zones Out-of-range (>25 cm), In-range (8-25 cm), and Close (<8 cm). Several example applications using the device's novel features were presented. These included a voice recorder triggered by holding the device up to one's head, an function that automatically wakes up the device when it is picked up, a technique that automatically changes the display orientation between portrait and landscape based on the orientation of the device, and lastly a tilt-to-scroll technique.

3.1.5 Prototypes and Systems in the Thesis

This last introductory section will briefly describe the prototypes and systems that were developed for this thesis. While each of the systems is presented in detail in their own chapters, a short summary of each system here functions as a reminder of the main features.

DiffDisplays (described in Chapter 7, also see Figure 3.2) is a prototype system used to evaluate visualisation techniques that show changes during periods of inattention. The main purpose of the techniques is to visually convey change that occurred while a person's attention was directed

[1] http://www.billbuxton.com/hydraNarrative.htm

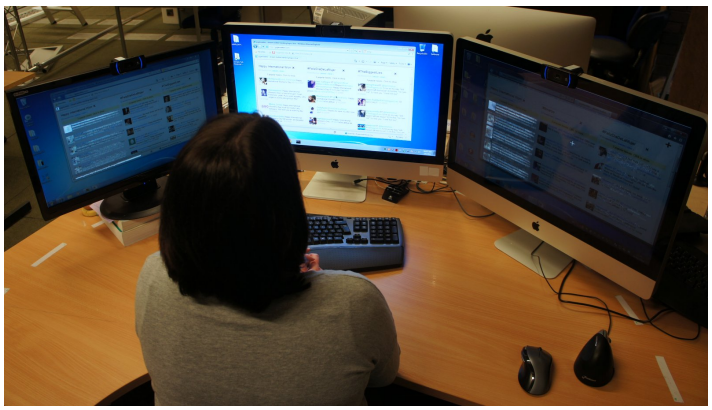


Figure 3.2: An Illustration of a system from this thesis. *DiffDisplays* tracks visual change on unattended displays through visualisations.

away from a display. Technically, the system forms a single or multi-display environment with each display being augmented by a web camera. The camera performs computer vision tracking of the person's eyes, which helps estimate whether they are looking in the direction of a specific display or not. When the person is looking at a display, the system lies dormant. After their attention shifts elsewhere, the system activates a visualisation technique, which tracks any visual changes on the unattended display. When the person's attention returns to the display, the visualisation slowly fades into the live state of the desktop, giving the person a chance to quickly establish regions where change has occurred.

MultiView Train Board (presented in Chapter 5) is a prototype system that exploits inherent properties of a specific type of an LCD display to show a distance and orientation dependent multiple simultaneous views of information. The display has three distinct views with the first two being shown based on how close the person interacting is to the display and the third view is shown simultaneously (with either the first or the second view) to observers far away from the display. The third view exploits a viewing angle specific behaviour of TN LCD displays[2], which allows to selectively show or hide content visible from certain angles.

MultiView Video Player (also described in Chapter 5) is another system that uses a combination of vision based tracking and inherent properties of TN LCD displays to show distinct simultaneous views for a film. People sitting on one side of the display see a video with subtitles, whereas people siting on the other side of the display only see the video and no subtitles. The size of the font for the subtitles is based on the distance of the viewers on the side of the display where subtitles are visible.

SpiderEyes (presented in Chapter 6) is an application system and part of a toolkit for developing spatially aware visual interfaces. By default, it comes with several example spatial techniques for manipulating visual on-display content. These techniques form the basis of this analysis. The size of the displayed content can be changed based on the distance to the display (magnification). The detail granularity can be altered (semantic zoom) or changed altogether through the use of different visual representations or the use of multiple datasets. The position along the display can determine either the positioning of the user's viewport/lens (in multi-user scenarios) or can be used as an input into a visualisation (in one example to literally walk through time). Orientation towards the display is used to filter out passers by. Distance to other users is used for determining grouping.

Now that all the systems included in the analysis have been introduced, the focus can shift to the actual analysis. The next four sections present the results of the analysis of spatial relationships

[2]Twisted nematic LCD — a type of LCD display known for its limited viewing angles and colour and contrast shifts at sub-optimal viewing angles

	Distance					
Name	AA	AO	AE	OO	OE	EE
Opo [HK+14]	C					
Puppeteer Display [BB+14]						
GroupTogether [MHG12]		C				
Is Anyone Looking? [BL+14]	C	C				
ProximityTable [Aln15; HR+14]	D	C				
LightSpace [WB10]	C	C				
Proxemic Media Player [BMG10]		C		C		
Hello.Wall [SP+03; PR+03]		D		D		
Public Ambient Displays [VB04]		D				
Proxemic InfoVis [JS+13]		C				
Medusa [AG+11]		C				
Proxemic Peddler [WBG12; Wan12]	C					
Mobile Sensing Techniques [HP+00]						
Vision Kiosk [CA00]		D				
Screenfinity [SMB13]		C				
E-conic [NS+07]		C				
EasyLiving [BM+00; KH+00; BS09]		C				
Active Hydra [Gre99; KG99]		D				
Range Whiteboard [JLK08]		D				
Authenticated Badge [WH92; WH+92]	P	D	P		P	
Lean & Zoom [HD08]		D				
Shadow Reaching [STB07]						
Psychic Space/Maze [Kru77; KGH85]						
Proxemic Controls [Led14; LGB15; GL13]				C		
Egocentric Spatial ZUI [RJ+13]				C		
Code Space [BD11]						
Touch Projector [BB+10]				C		
Context-Aware Applications [SAW94]	C	C	P			
Sentient Computing System [AC+01; HH+02]		C	P			
DiffDisplays		P				
MultiView Train Board		D				
MultiView Video Player		C				
SpiderEyes	C	C				

Table 3.1: Relationship - Distance

Relationship legend: A - Actor, O - Object, E - Environment; e.g. AO — Actor-Object
Value legend: C - Continuous adaptation; D - Discrete, zone or threshold based adaptation; P - Binary adaptation based on presence/absence

between entities within each of the systems, followed by a discussion on both main results and observations that arose during the analysis, as well as limitations of the analysis. The first relationship that is examined is *distance*.

3.2 Relationships - Distance

Distance is by far the most common spatial relationship used by systems. Table 3.1 summarises the values for each system. Distance is also a spatial relationship with the highest portion of systems mainly making use of a single entity relationship, namely Actor-Object distance. However, with the

exception of the Environment-Environment relationship, all other entity combinations have at least one system making use of the relationship.

3.2.1 Actor - Actor

Interestingly, even though a number of authors used proxemics as the guiding idea behind the design of their systems, only a small portion (8) of the analysed systems actually use interpersonal, or more accurately Actor-Actor, distance for interactional purposes. The relationship is used in a fairly uniform manner between systems, primarily to establish or formalise a relationship between two (or more) Actors. This is the case for *SpiderEyes* and *ProximityTable*. Both systems use interpersonal distance to determine when a group is formed.

With *Proxemic Peddler* the distance between a person and a display also plays an important role. However, with *Proxemic Peddler* both the person and the display are Actors as the display attempts to manipulate the person first into interacting with the display and eventually into making a purchase. Therefore, even though it is the distance between a person and a display, within the IRE model an Actor-Actor distance relationship is formed.

Context-Aware Applications, *LightSpace* and *Opo* use Actor-Actor distance to establish relationships. With *Context-Aware Applications*, distance is used to determine which other Actor is the target of an information transfer. Similarly, *LightSpace* uses distances of the people's meshes to ascertain when physical contact is established and therefore when to trigger a transfer of information. With *Opo*, the distance between people is used to determine when face-to-face interaction with another person takes place.

Lastly, there is *Authenticated Badge* and *Is Anyone Looking?*, which use Actor-Actor distance in arguably the least noticeable ways. With *Authenticated Badge*, people can query the system to find out how many (and which) other individuals are in a particular person's proximity. In *Is Anyone Looking?*, Actor-Actor distance is used to calculate the width of a silhouette of a passer by who may be a potential shoulder-surfing attacker. This reverse-shadow is used to obscure potentially sensitive information on the display to protect the privacy of the primary user.

3.2.2 Actor - Object

The Actor-Object distance is by far the most common distance relationship. It is used by 24 of the systems. This comes as little surprise as most of the spatially-aware system contain an interactive display. The use of Actor-Object distance generally falls into one of three categories - information targeting, interaction engagement, or visual changes.

Information Targeting Systems in this category use distance between an Actor and an Object to form a relationship between the two to either directly transmit information or to identify, which Actor is the source of input into the system. A number of the smart space and smart room systems make use of this relationship. In the *Sentient Computing System* and the *Authenticated Badge*, the telephone closest to a certain person is used as a target device when a call to that person is made. Additionally with the *Authenticated Badge*, being within a certain distance of a workstation triggers an identity validation protocol. Both of these behaviours are also present in the *Context-Aware Applications* system, except there they are used as a more general targeting strategy for selecting resources (e.g. printers and other devices) or for triggering actions. *LightSpace* also uses the Actor-Object relationship to trigger actions but in this case it is to trigger a transfer of a digital object when a person makes contact with a surface. Additionally, *LightSpace* uses distance from a point on an interactive surface (e.g. a table or the floor) to navigate a vertically arranged spatial menu. *Medusa* uses a similar technique for hover interaction, where the distance of a person's arm to a tabletop display is used to spatially arrange element/visual layers in their UI prototyping application.

In the *EasyLiving* system, keyboard input is assumed to come from the person closest to the keyboard while the keyboard input was generated. *GroupTogether* also uses distance between devices and people to form a link between them where each device is matched with a person by assigning them to the person closest to the device (but at most one metre from the device).

Interaction Engagement One of the most common uses for the Actor-Object distance relationship is to gauge how engaged the Actor is with the Object. *Vision Kiosk* uses the distance of a person from the kiosk to decide when the on-screen avatar should attempt to initiate a conversation with the approaching person. Additionally, the avatar will always face the person closest to the Kiosk, the assumption being that the person closest to an interactive display drives the interaction. Their distance is used to determine, which interaction zone they are in. This also used in *Range Whiteboard* and *MultiView Train Board*. In *Range Whiteboard*, *MultiView Train Board*, together with systems supporting simultaneous parallel use by multiple people, namely *Public Ambient Displays* and *Hello.Wall*, the Actor-Object distance enables or disables certain types of interactions, depending on which interactive zone each of the Actors is in. With *Public Ambient Displays*, other triggers are used for some of the interaction zones but distance is used to transition into the subtle interaction zone, which reveals more detail about the available information.

ProximityTable, *Hello.Wall*, *DiffDisplays*, and *Medusa* use the presence or absence of people within the sensor range. In the case of *ProximityTable*, the assumption is that as long as the person is within sensor range, they are interested in interacting with the tabletop. *Hello.Wall* is even more explicit, where if no people are present in the main display's sensor range, it shows general information or aesthetically pleasing light patterns, but as soon as someone enters sensor range, they are in the notification zone and information for them is shown. *DiffDisplays* approaches the relationship from the opposite end. Sensor range (as a measure of distance) is used as an additional constraint for determining attention towards displays. The system was designed so that if a person is beyond the sensor range, they are highly unlikely to be able to read any content on the display and therefore it can be safely assumed that they are not paying direct attention to the display. *Medusa* also uses the presence or absence of people within the sensor range. Unlike *ProximityTable* and *Hello.Wall*, *Medusa* does not automatically assume that people want to interact when in sensor range. It does however, try to convey that it detects them and their behaviour by displaying a "user orb". The orb changes its blurriness based on the distance of the person to the tabletop display.

Active Hydra is the last system to use the Actor-Object distance to gauge engagement. Within the system, distance to the Hydra device is used to infer how much a person wants to interact with their remote partner. The distance from the Object limits the quality of the audio-video link. At close distances the full audio-video capability is enabled. As one of the partners moves away from the Hydra, first the audio is disabled and then video switches to much slower frame rate.

Visual Changes The systems mentioned here use the Actor-Object distance to perform changes to the visual content shown on displays. *Is Anyone Looking?* uses a number of indicators to communicate the distance of a potential shoulder-surfer to a display. In one case, the distance of the shoulder-surfer determines the transparency of a flashing border around the display. In another of the privacy enhancing techniques, the distance to the display is indicated by changing the size of a 3D model of a human body symbolising the shoulder-surfer. *MultiView Video Player* uses the distance of people from the display to keep the apparent size of subtitles the same regardless of the person's distance to the display. *Screenfinity* and *E-conic* also use distance for the same purpose, keeping the size of the displayed content the same apparent size. The *Proxemic Media Player* uses changes in distance between a person and a display to demagnify content on the display and to change its layout to provide more detail as the person approaches. *Lean & Zoom* uses distance to magnify content. Instead

of keeping the apparent size constant, the system amplifies the movement of the tracked person. The system also has an application that uses semantic zoom rather than visual zoom.

Proxemic InfoVis employs the Actor-Object distance for a different purpose in each of the three studies performed. In Study 1, it is used for amplifying magnification similarly to *Lean & Zoom*. In Study 2, distance to the display changes the level of granularity for visualisations. Study 3 uses distance to select which visualisation attribute is examined. *SpiderEyes* combines all the previous functionality together. Distance of a person to a display can be used for any of the two magnification types (amplifying magnification with movement or keeping apparent size constant), it can be used for aggregating data in visualisations (or for providing more data granularity), or it can be used to change visualisation types of datasets altogether.

3.2.3 Actor - Environment

Only three systems make use of the Actor-Environment distance relationship. The discussion at the end of the chapter includes an examination of some of the reasons for this, and some of the complexities of analysing relationships with Environment entities in general. All of the three systems fit into the smart space theme. *Authenticated Badge* uses the relationship only in a very limited fashion by allowing people to query the system to find out, who is present in a particular room. In the *Sentient Computing System*, absence from and presence in a person's own office triggers call forwarding to the closest telephone on and off, respectively. *Context-Aware Systems* generalises the pattern by allowing people to trigger arbitrary system action on entering or leaving Environments.

3.2.4 Object - Object

Object-Object distance is relatively infrequently used with only five systems actively using it. *Proxemic Controls* makes the most extensive use of the relationship. Distance of the universal controller (a tablet) to an appliance is used as part of a gradual engagement pattern to determine how much information about the appliance is visible, and how much control over the appliance can be exerted using the controller. The Object-Object distance is also used to determine an orientation threshold, within which the controller is interpreted as interacting with a particular appliance (how accurately the tablet needs to point towards the appliance).

Touch Projector and *Egocentric ZUI* both use the distance of a mobile device to a remote display to control content magnification. *Touch Projector* attempts to keep the apparent size of items on the mobile display constant, while *Egocentric ZUI* amplifies the magnification due to movement. The *Proxemic Media Player* uses the distance from a portable media player to a shared large display to display differing levels of granularity of content on the media player that can be shared to the large display. Very short distance between the media player and the display allowed content to be transferred onto the shared display. Similarly, *Hello.Wall* uses inter-Object distance to enable content transfer. When a ViewPort is within the cell-interaction zone (very close to the public display), transfer of information from individual display cells to the ViewPort is enabled.

3.2.5 Object - Environment

Only one system, *Authenticated Badge* uses the Object-Environment distance relationship and only in the presence/absence form. Presence of portable workstations (laptops) is tracked so that they can be unlocked or locked when their owner enters or leaves the room, respectively.

3.2.6 Environment - Environment

The Environment-Environment relationship is not used by any of the analysed systems.

	Position					
Name	AA	AO	AE	OO	OE	EE
Opo [HK+14]						
Puppeteer Display [BB+14]	C					
GroupTogether [MHG12]	C	C	C	D	C	
Is Anyone Looking? [BL+14]	C	C				
ProximityTable [Aln15; HR+14]		C				
LightSpace [WB10]						
Proxemic Media Player [BMG10]	C	C	P	C		
Hello.Wall [SP+03; PR+03]				D		
Public Ambient Displays [VB04]		C				
Proxemic InfoVis [JS+13]		C				
Medusa [AG+11]						
Proxemic Peddler [WBG12; Wan12]						
Mobile Sensing Techniques [HP+00]		D				
Vision Kiosk [CA00]						
Screenfinity [SMB13]		C				
E-conic [NS+07]		C		C		
EasyLiving [BM+00; KH+00; BS09]		C				
Active Hydra [Gre99; KG99]				D		
Range Whiteboard [JLK08]						
Authenticated Badge [WH92; WH+92]		P	P			
Lean & Zoom [HD08]						
Shadow Reaching [STB07]		C				
Psychic Space/Maze [Kru77; KGH85]			C			
Proxemic Controls [Led14; LGB15; GL13]					P	
Egocentric Spatial ZUI [RJ+13]				C		
Code Space [BD11]		C				
Touch Projector [BB+10]				C		
Context-Aware Applications [SAW94]						
Sentient Computing System [AC+01; HH+02]		C	C			
DiffDisplays						
MultiView Train Board						
MultiView Video Player						
SpiderEyes	(C)	C	(C)			

Table 3.2: Relationships - Position

Relationship legend: A - Actor, O - Object, E - Environment; e.g. AE — Actor-Environment
Value legend: C - Continuous adaptation; D - Discrete, zone or threshold based adaptation; P - Binary adaptation based on presence/absence. Values in brackets are not used in the example applications provided but are available as part of the *SpiderEyes* toolkit.

3.3 Relationships - Position

Position relationships are the second most common spatial entity relationships in use by the analysed systems. Table 3.2 summarises the values for each system. It should be noted that position relationships capture two kinds of spatial relationships. The first is for spatial relationships that cannot be described by either distance or orientation (e.g. position along a line parallel to a display), either alone or in combination. The second is when the relationship could be described using distance and orientation but the system specifically uses the combination of distance and orientation

together to arrive at a value. Similarly to distance, all entity combinations have at least a single system using the spatial relationship for interaction.

3.3.1 Actor - Actor

Once again only a relatively small number of systems make use of the Actor-Actor relationship. *Is Anyone Looking?* and *Proxemic Media Player* use positions in front of display. In *Is Anyone Looking?*, the positions of persons in front of the display are used to cast a reverse shadow on the display, where the aim of the shadow is to block the visibility of potentially sensitive content from the person behind the primary user as they are seen as a shoulder-surfing attacker. *Proxemic Media Player* uses the positions of persons in front of its shared display to determine the sharing strategy for the display. If both persons can see the display, the display shows the content belonging to both of them. However, if one person significantly occludes the view of the other person (essentially standing between the second person and the display), only their content is visible. *Puppeteer Display* uses the position between people to firstly determine whether they are too close to each other and if they are to attempt to distribute them along the display more evenly so they do not interfere with each other.[3]

GroupTogether uses a combination of distance and orientation to determine when a group is formed. Two or more persons need to be within a certain distance and facing towards the group centre for a group to be formed. This is made somewhat complex by the fact that a group also corresponds to the interactive environment, so interactive environments are formed dynamically based on this relationship. To complicate matters further, a shared display (Object) can also be part of the group, so it is the combination of the Actor-Actor and Actor-Object relationship that is used to form groups.

3.3.2 Actor - Object

As with the distance relationships, Actor-Object is also the most frequently used position relationship with 16 systems making active use of it. By far the most common application of the relationship was to use it to position elements on a display. *Screenfinity* uses the position of people along a wall display to translate the position of content on the display so that it follows the reader to maintain its readability for longer. When using *ProximityTable*, workspaces also follow people as they move along the side of the tabletop display. The same is also the case with *Ambient Public Displays*, *Proxemic InfoVis*, and *SpiderEyes*. However, *Proxemic InfoVis* and *SpiderEyes* can also use the position along the display to navigate a dataset, with *Proxemic InfoVis* also supporting panning and menu selection. A somewhat related technique is used in *Shadow Reaching*, where in one prototype, the shape and position of a virtual shadow of people using the display was determined by their position. More interestingly, this shadow was used as a lens to provide an alternative view of the displayed data. In one other prototype, the position of a person in relation to a physical (or virtual) light source was used to extend their reach when pointing on a remote display.

In *Proxemic Media Player*, the Actor-Object position is used for several techniques. Position of persons relative to a shared display is used to determine which side of the display their content is shown on, while the display is used simultaneously by two people. The paper also mentions that the position of a person relative to a display is used to decide whether they want to browse videos or whether they want to watch the currently playing video (but the description of the technique is unclear). Lastly, the relative position of a person's head and a mobile phone is used to determine whether the mobile phone is used as a pointer or not. Pointing behaviour is also part of a technique implemented in the *Code Space* system. If a person points a mobile phone at a display while holding is

[3]The cited paper [BB+14] does not make it clear whether this is just horizontal distance between them or not and uses the term position exclusively in the description, so the relationship is interpreted as a position relationship.

perpendicular to the ground at arm's length from their body, the contents of the phone's display will be temporarily shown on the display that is being pointed at. This kind of complex positioning of a mobile device is also used in *Mobile Sensing Techniques* for two techniques. Holding the augmented mobile device at close proximity and upright will trigger voice recording, which will stay active until the device is moved more than 25 cm from the person. Another technique uses a combination of distance and orientation to wake up the device and prevent it from going back to a sleep state.

Authenticated Badge, *Sentient Computing System*, and *EasyLiving* all use the Actor-Object position to track some form of containment. In one of the example applications for *Authenticated Badge*, one of the LEDs on the badge will light up whenever the holder of the badge is in the field of view of a camera. *EasyLiving* seems to use a combination of distance, line-of-sight and other constraints to determine which display a person's desktop session should be directed to. The *Sentient Computing System* also directs a person's desktop session to a specific workstation, but here the mechanism is based on containment within one of two zones that extend around instrumented workstations. Presence in the (smaller) active zone triggers the transfer of a person's desktop session to the workstation, and for as long as the person is present in the (larger) maintenance zone, the session will remain on the workstation.

The last three systems to use Actor-Object position are *GroupTogether*, *Is Anyone Looking?* and *E-conic*. The large shared display available in the *GroupTogether* interactive space is not an Actor in the IRE model sense but it is interpreted as an entity capable of forming a part of an interactive group in the paper. As the shared wall display can also form part of a group, the system also uses a combination of the display's distance and orientation to other Actors (and vice versa) to determine whether a group should be formed. In one of the privacy-enhancing techniques, *Is Anyone Looking?* highlights a border of the public display to indicate which direction a potential shoulder-surfer is approaching from. Another technique positions a simulated body model along the display to indicate the position (and orientation) of the shoulder-surfer. Approximate gaze direction can also be shown on the display. *E-conic* uses the people's position to enable it to render a virtual canvas so that windows can seamlessly (for a particular person) span multiple displays or provide a context+detail view.

3.3.3 Actor - Environment

Four systems make active use of an Actor's position within an Environment, though for the first two it is mainly in terms of presence. *Proxemic Media Player* uses a person's presence at the entry point of a room to define a trigger point for turning the system on or off. *Authenticated Badge* uses a person's entry into a room to lock or unlock computers that belong to them.[4] The *Sentient Computing System* can track a person's position within the Environment to produce a video that keeps the tracked person always in the video frame by switching between available cameras.

GroupTogether is a room scale system that centres around small group interaction. In this system, Environments correspond to groups of people rather than rooms. The distance of individuals to a group as well as their orientation towards it are used to determine if they are part of the group, or a passer by. The last system to use the Actor-Environment position is *Psychic Space/Maze*, which maps the position of a person within a room to a position within an on-screen maze.

[4]This is very similar to a technique in the *Sentient Computing System*, which used presence in one's own office to disable call forwarding. The reason why the technique in *Sentient Computing System* is classes as a distance relationship, while the one in *Authenticated Badge* is classed as position-based is due to the latter's explicit use of the door as the location where the unlocking action would happen.

3.3.4 Object - Object

The second most commonly used position relationship was the Object-Object relationship with six systems exploiting it in some way. *Proxemic Media Player* uses the position of a media player in relation to a large shared display to determine where along the display the icons representing the media player's content should be shown. *GroupTogether* uses position to perform canvas merging — when two devices are side-by-side and at similar orientations, the canvases associated with the devices are merged, which enables the use of a cross-device pinch-to-zoom technique. *Egocentric ZUI* uses the position of a tablet in relation to large display to determine which part of the interactive canvas is shown on the tablet, essentially transforming it into a lens that offers a different view of the content.

Touch Projector uses the position of a mobile phone in relation to another display to translate touch events from the mobile device to the target display. This is only a small part of a more complex technique. *Hello.Wall* uses position of the ViewPort device in relation to the cells on the shared display to determine which cell of the display is being interacted with so that correct information can be transmitted to the device. Lastly, *Active Hydra* uses the position of a responding surrogate in relation to its stage to determine a person's openness to communicate. When the figurine is on its stage, it is interpreted as the person being interested in communicating with the remote collaborator. If it is not on the stage, they are assumed to be open to communication. A tipped over figurine means that the person has no interest in communicating.

3.3.5 Object - Environment

Only two systems make any use of the Object-Environment position. For *Proxemic Controls*, it is only in its presence/absence form. The system shows an overview of all available appliances within a room when the universal controller tablet is enters the room through the door. At any other position in the room, the standard control interface is shown. *GroupTogether* was described above as using Actor-Actor and Actor-Object distance and orientation to form interaction groups (Environment). So, analogously to the Actor-Environment relationship, *GroupTogether* uses distance and orientation or the shared display to the group to determine whether it is part of the group or not. Additionally, *GroupTogether* uses the presence or absence of devices within the group's o-space (the area in front of and between group members) to enable the majority of the systems content sharing and manipulation techniques.

3.3.6 Environment - Environment

No system in selected for analysis utilised this relationship.

3.4 Relationships - Orientation

The orientation relationship is used by systems less frequently than other relationship types. Table 3.3 shows the values for each system. Moreover, fewer combinations of entities are used by at least one system, with five out of nine combinations being used. There also seems to be less variety in terms of interaction techniques, with most common uses being view manipulation and as a proxy for visual attention.

3.4.1 Actor - Actor

Four systems made active use of the Actor-Actor orientation. With *Puppeteer Display* and *Proxemic Peddler* it is actually the orientation of a person towards a display that is tracked. However, in both of

	Orientation								
Name	AA	AO	AE	OO	OA	OE	EE	EA	EO
Opo [HK+14]	D								
Puppeteer Display [BB+14]	D								
GroupTogether [MHG12]				D					
Is Anyone Looking? [BL+14]		C							
ProximityTable [Aln15; HR+14]					D				
LightSpace [WB10]					C				
Proxemic Media Player [BMG10]	C	C							
Hello.Wall [SP+03; PR+03]									
Public Ambient Displays [VB04]		C							
Proxemic InfoVis [JS+13]		C							
Medusa [AG+11]					D				
Proxemic Peddler [WBG12; Wan12]	C								
Mobile Sensing Techniques [HP+00]									
Vision Kiosk [CA00]					C				
Screenfinity [SMB13]					C				
E-Conic [NS+07]					C				
EasyLiving [BM+00; KH+00; BS09]									
Active Hydra [Gre99; KG99]						C			
Range Whiteboard [JLK08]									
Authenticated Badge [WH92; WH+92]									
Lean & Zoom [HD08]									
Shadow Reaching [STB07]									
Psychic Space/Maze [Kru77; KGH85]									
Proxemic Controls [Led14; LGB15; GL13]				C					
Egocentric Spatial ZUI [RJ+13]									
Code Space [BD11]		C							
Touch Projector [BB+10]									
Context-Aware Applications [SAW94]									
Sentient Computing System [AC+01; HH+02]					C				
DiffDisplays		D							
MultiView Train Board					D				
MultiView Video Player					D				
SpiderEyes	(C)	C			(C)				

Table 3.3: Relationships - Orientation

Relationship legend: A - Actor, O - Object, E - Environment; e.g. AO — Actor-Object
Value legend: C - Continuous adaptation; D - Discrete, zone or threshold based adaptation. Values in brackets are not used in the example applications provided but are available as part of the toolkit.

these systems the displays play an active part in the interaction and try to influence the behaviour of the people interacting, and so the displays are judged to be Actors in the interaction. The *Puppeteer Display* uses a simple binary notion of orientation to determine whether a person is facing the display or not. This is used to evaluate the effectiveness of techniques for attracting attention of passers by. *Proxemic Peddler* uses orientation as an indicator of engagement. When a person's orientation intersects with the display, the display tries to entice them to start interacting by showing them a product. If the person turns away, a different product is shown in an attempt to lure them back. Movements of content on the display is used to increase the chances that the actions taking place on the display will be noticed.

Proxemic Media Player uses the Actor-Actor orientation to determine whether they are engaged in a conversation, in which case it pauses any video playback. *Opo* only uses orientation marginally, both for technical reasons and by design. The system only collects the orientation data. However, even then the collected data is binary in nature as the system is only able to determine if a person is approximately facing in the direction of another instrumented person or not.

3.4.2 Actor - Object

The Actor-Object orientation is the second most frequently used orientation relationship after Object-Actor orientation. By far the most common use of the Actor-Object orientation was as a proxy for gaze direction or visual attention, generally towards a display. In all cases, this was partly due to the tracking technologies used as all of the remaining systems primarily tracked the head or part of the head. *Is Anyone Looking?* used the orientation of the person further away from the display to determine whether they are looking in the direction of the display and, by extension, whether they were a passer by or a potential shoulder-surfing attacker. If they were judged a potential attacker, one of the display's privacy enhancing techniques would be triggered. *SpiderEyes* used the relationship in a very similar manner. If a person did not look in the direction of the display, they would be judged as a passer by. However, as soon as they turned towards the display, they would become treated by the system as an active user. *Public Ambient Displays* also used a person's orientation to filter out passers by. However, if a passer by looked towards the display it would cause the display to show a more detailed view of information available for them. Additionally, the body orientation would determine the width of their interactive on-screen representation, while the head orientation would determine its transparency.

In *Proxemic Media Player*, orientation of persons towards Objects was used for two variations of a technique. In the first variation, when the person's head was not pointed in the direction of the display, video playback was paused. This technique was later extended because it was deemed too inaccurate (while the person may not be currently looking at the display it does not mean they wish the video to be paused as they may only be looking for their drink). In the second, extended, version, the Object towards which attention was directed was the determinant. If the person was looking towards a newspaper, playback would be paused. However, if they were looking at a bowl of popcorn, playback would continue.[5] *Code Space* used orientation to determine when to switch the display from an audience oriented mode, when the presenter in front of a shared display was not looking at the display, and a presenter mode, which showed special menus to help the presenter with the presentation. In *DiffDisplays*, orientation was also used as a proxy for attention, tracking which display(s) a person was looking at or not. However, the system concentrated on inattention, where as soon as the person was not looking at a specific display, the system would trigger a technique visualising any change on the display that took place before the tracked person looked at the display again.

The only system not using Actor-Object orientation as a proxy for visual attention or gaze direction was *Proxemic InfoVis*. In study 1, orientation of a person's head was used for up/down panning of the display content.

3.4.3 Actor - Environment

The Actor-Environment relationship was not used by any of the system in this analysis.

[5] It was not clear from the description to which level this technique was actually implemented.

3.4.4 Object - Object

Two systems utilise Object-Object orientation for interactive purposes. *GroupTogether* implements a technique called Portals, which links two tablets together when one of them is tilted towards the other. On tilt, both tablets show a tint along the corresponding edges of the tablet displays and if one of the Actors drags any content through the edge or one of the tablets, the content is transferred to the other tablet. *Proxemic Controls* explores both directions of the Object-Object orientation relationship between the universal controller tablet and an appliance. The orientation of the universal controller towards an appliance determines whether the controller can exert any control over the appliance (or which of a set of appliances is being controlled). Using the reverse relationship, the orientation of the appliance in relation to the controller can be used to control specific aspects of the appliance. Taking a radio as an example, pointing the tablet towards it allows the person holding the tablet to view some of the radio controls. However, positioning the tablet such that the tablet is to the left of the radio (the radio is facing to the left of the tablet) reveals volume controls, where placing it above the radio (the radio is facing the space underneath the controller) reveals controls for switching stations.

3.4.5 Object - Actor

Similarly to distance and position relationships, the orientation Object-Actor relationship is also the most commonly used one, with eight systems making use of it. Most often, the relationship was used to optimise the viewing of content on a display for a particular Actor. In the case of *LightSpace*, the orientation of a spatial menu towards a person was used to determine the orientation of text so that it would always remain easily readable. *Screenfinity* has an identical technique, except unlike *LightSpace* where the projected surface was horizontal, here the display surface is a large vertical display. The rotation is also performed continuously. *E-conic* goes one step further by performing full perspective correction to keep the appearance constant from a particular person's point of view.

ProximityTable and *Medusa* both use the orientation of the tabletop display. In *ProximityTable*, if a person moves from one side of the tabletop to another, their workspace is re-oriented so it faces the person. *Medusa* uses the tabletop sides as a mode switch where some functionality is only available when the person faces a specific side of the tabletop. *MultiView Train Board* and *MultiView Video Player* also use the physical properties of the interactive devices to determine views, although in their case it is exploiting the multi-view capability of the employed display. In *MultiView Train Board* a person's orientation to the display along the vertical axis determines whether they will see one of the more detailed views, which are distance dependent, or whether they will see the distance independent general content. In *MultiView Video Player* the angle from which the display is viewed determines whether a person will see a subtitle track or not.

The last two systems to use Object-Actor orientation are *Vision Kiosk* and *Sentient Computing System*. In *Vision Kiosk*, the avatar's face that communicates with a person always faces towards them based on the kiosk's orientation to the person. *Sentient Computing System* uses orientation of the camera to determine who is in the picture, when a picture is taken with a fix-position camera.

3.4.6 Object - Environment

The last orientation relationship that at least one system actively used was Object-Environment orientation. *Active Hydra* uses the orientation of the peek-a-boo surrogate to indicate the activity level of a remote person. The surrogate facing the wall indicates that the remote person is not present. When the surrogate faces the room, the remote person is active within their office.

3.4.7 Environment - *

None of the following relationships were used by any system — Environment-Environment, Environment-Actor, Environment-Object.

3.5 Relationships - Discussion and Limitations

A number of observations and potential limitations arose during the analysis. This section addresses observations about the analysis itself and some limitations relating to the classification of systems with specific focus on entity relationships. Discussion on the analysis results, gaps and trends is in Section 3.11.

Classification as Actor-Environment/Object-Environment or Actor-Object/Object-Object Relationship Some of the systems use the presence or absence (or the act of entering or leaving) as actionable events. What may be somewhat confusing is why techniques using these events are sometimes classified as relationships with Objects and sometimes as relationships with Environments. While the state of the Object or action of the Actor may be more or less identical in all cases, and they may even lead to the same result, the classifications differ. The determining factor is the description of the systems. With some of them it is clear that all the Actors and Objects interact in an Environment of some description. For those systems, the actions and relationships were classed as Actor-Environment or Object-Environment. Where all interactions are described only from the perspective of the Actors or Objects and the concept of an Environment is never discussed, the relationships were classed as Actor-Object or Object-Object as it was impossible to determine whether the systems have any notion of Environment (this is specifically the case with *ProximityTable*). This decision affected the classification of the following systems: *GroupTogether*, *Proxemic Media Player*, *Authenticated Badge*, *Context-Aware Applications*, *Sentient Computing System*, *ProximityTable*, *Hello.Wall*, *DiffDisplays*, and *Medusa*).

Another complication arose due to some systems using a relationship with the ground, namely *LightSpace* where distance from a point on the ground or table or other surfaces is used to select menu items. From the description is it unclear whether it should be considered an Actor-Environment or Actor-Object relationship because it is not clear whether the ground is considered to a part of an environment or whether it forms an object that can be interacted with. Within this analysis, the relationship is interpreted as Actor-Object because the intent is interaction with the spatial menu. The menu is essentially an interactive object, which happens to have no tangible representation. The description uses the word floor to clearly convey the plane from which the distance is measured but the actual interactive object is a point placed on the floor (or another surface).

Interactive Extents of Objects as Opposed to Environments Some systems, namely *Authenticated Badge*, *Sentient Computing System*, *EasyLiving*, and *Context-Sensitive Applications*, use regions around objects to trigger actions such as transferring a person's desktop session to a workstation. This can be interpreted in two ways, the interaction is a customisation of the Object, in which case the relationship is Actor-Object (this is usually a position relationship as it is often not a circular region, so simple distance does not work). However, it could also be interpreted as a personalisation of a workspace environment, in which case the relationship would be presence/absence within an Environment also containing the interactive Object. While that seems to be the correct interpretation in terms of the interaction (moving into the workspace environment conveys intention to engage in a workspace interaction scenario, which is facilitated by customising the objects in the environment), since the publication authors concentrated on the objects themselves (including the way supporting

infrastructure was described, e.g. implementation and middleware layers), the analysis generally uses the Actor-Object relationship as that seems to have been the intention of the system authors.

A counter example could be *Psychic Space/Maze* where the description implies the position relationship with a room (Environment) is tracked and used but the system would likely work equally well if the relationship was with a display (it is likely that the description and hence the interpretation was influenced by the fact that the entire floor of the room was covered with the pressure sensitive tiles).

Position as Distance and Orientation Combined Some systems use the combination of distance and orientation to trigger an action, for example *Mobile Sensing Techniques* uses a combination of distance (<8 cm from a person) and orientation (device is held vertically) to trigger a voice recording. While parts of the technique use simple distance and parts use orientation, it is the combination of the two constraints that is the trigger. Therefore, the spatial relationship cannot be described by either distance or orientation alone and so it falls into the position category.

Entities and Their Parts One of the complications that arose during the analysis was determining the level of granularity at which a relationship is formed. With Objects, this was relatively straightforward as when their spatial information was used, it tended to be their overall state that was used regardless of where the measurement was taken (spatial centre, surface, etc.). With humans the situation became more complex as while some systems considered the body as a whole, most systems only tracked some part(s) of the human body. Most commonly this was the head or the torso. Some systems also tracked the arms/hands/fingers to enable gesture interaction.

This presents a problem because for example pointing is an action, which by definition requires the use of spatial information so that the target of the gesture can be established. However, considering these types of actions within the analysis as spatial interactions significantly complicates the analysis. Since one could argue that, for example, using a mouse also constitutes spatial interaction as the device translates the spatial coordinates of the hand into two dimensional workspace coordinates. The same applies to most other pointing devices and to arm/hand/finger pointing gestures.

In order to address this issue, an additional constraint for inclusion in the analysis is imposed. Where people form part of the interaction, only systems where the majority of the body is used for the interaction technique are included in the analysis and any interaction based solely on arm, hand or finger gestures is excluded from the analysis. Using a body part (such as a head, torso, etc.) as a proxy for the entire entity does not exclude the system from analysis (even in cases where mostly arms/hands/fingers are tracked as is the case with *Medusa*).

For interaction using devices, systems where only a mouse or a similar standard pointing device is used in a spatial manner are not included in the analysis. Moreover, systems where the device involved only functions as a direct analog to a mouse or a similar pointing device are also not included in the analysis. However, where other properties and characteristics of the device are used to enhance spatial interaction (as is the case with *Touch Projector*) or where the pointing device is clearly used as a proxy to enable the performance of a more complex spatial interaction technique (as with one of the prototypes in *Shadow Reaching*), the system is included in the analysis.

Classifying Pointing and Selection Following directly on from the last discussion point, a clarification with regards to pointing and selection is needed. Both of these actions inherently involve spatial information as otherwise it is not possible to determine the target of the actions. However, for the purpose of this analysis, a distinction between different types of pointing and selection mechanisms has been made. With human interaction, where the interaction uses only a pointing/selection gesture

for targeting purposes, the specific interaction technique is not included in the analysis.[6] However, where the pointing/selection gesture only forms a part of a more complex technique using other spatial information, the technique was included in the analysis.[7]. In cases where the pointing or selection was performed by the entire body (or its proxy body part), the technique was included in the analysis.[8]

With primarily device-based interactions, as above, where the device was part of a more complex technique, or where device properties beyond its ability to be used as a targeting indicator were used, the interaction technique was not excluded from the analysis.[9] On the other hand, where the device or object was used solely as a directional or targeting indicator, the technique was excluded from the analysis.[10]

Single Entity Use This is a limitation of the current design of the IRE model. The model was designed to capture inter-Entity spatial relationships as they are used for interactive purposes and arguably the model performs very well in this regard. However, the model is currently limited in how well it can capture interaction techniques that only use the spatial properties of a single entity without any specific relationship to another entity. A good example of this construct is an interaction technique in *Mobile Sensing Techniques*, where the mobile device's orientation determines whether the display is in portrait or landscape mode. This technique clearly uses the spatial properties of the Object for interaction. However, this cannot be easily captured in the current model design as no relationship to any other entity is formed. Arguably this could be classified as a relationship with an Environment (with the environment being the world). Unfortunately, this is not very accurate as the technique could be equally well performed in any location (assuming the orientation sensors were location independent), regardless of any specific Environment entity.

This concludes the part of the analysis concentrating on spatial relationships between entities. Now the focus will shift towards analysing interaction properties, namely interaction range, entity cardinality, action intentionality, and action intensity.

3.6 Interaction - Range

Entity relationships are the main focus of the IRE model. However, an understanding of the analysed systems would be very incomplete without additional insight offered by examining their interaction characteristics. That said, the values for the interaction characteristics are much more self-explanatory than the nuanced use of spatial relationships. The following sections will provide a brief overview of the values and trends for each analysed interaction characteristic with more detailed descriptions where appropriate. However, since the descriptions and values will be very similar for many of the systems, details will be highlighted only in cases of particular interest.

Table 3.4 shows range values for all the analysed systems. Out of the 16 possible combinations of range values, only six were used by systems. Where multiple ranges were used, they formed a continuum (i.e. there were no gaps between ranges - for example intimate range and public range). In this section, range specific notes will be presented first, followed by an overview of the range value clusters.

[6] This decision lead to exclusion of one or more techniques from the following systems - *Code Space*, *Ambient Public Displays*, *Proxemic Media Player*.

[7] The lead to the inclusion of *Shadow Reaching* in the analysis.

[8] This was the case for *SpiderEyes*, *Psychic Space/Maze*, *Proxemic InfoVis*.

[9] This applied to a technique in *Touch Projector* and *Code Space*.

[10] This affected *Proxemic Media Player*.

Name	Range: intimate (<0.6m)	personal (<1.5m)	social (<5m)	public (>5m)
Opo [HK+14]	x	x	x	
Puppeteer Display [BB+14]		x	x	
GroupTogether [MHG12]	x	x	x	
Is Anyone Looking? [BL+14]		x	x	
ProximityTable [Aln15; HR+14]	x	x		
LightSpace [WB10]	x	x	x	
Proxemic Media Player [BMG10]	x	x	x	
Hello.Wall [SP+03; PR+03]	x	?	?	
Public Ambient Displays [VB04]	x	x	x	
Proxemic InfoVis [JS+13]		x	x	
Medusa [AG+11]	x	x	x	
Proxemic Peddler [WBG12; Wan12]	x	x	x	
Mobile Sensing Techniques [HP+00]	x			
Vision Kiosk [CA00]	x	?	?	
Screenfinity [SMB13]		x	x	x
E-conic [NS+07]	x	x	x	
EasyLiving [BM+00; KH+00; BS09]	?	?	?	
Active Hydra [Gre99; KG99]	x	?		
Range Whiteboard [JLK08]	x	x		
Authenticated Badge [WH92; WH+92]	x	x	x	x
Lean & Zoom [HD08]	x	x		
Shadow Reaching [STB07]		x	x	
Psychic Space/Maze [Kru77; KGH85]	x	x	x	x
Proxemic Controls [Led14; LGB15; GL13]	x	x	x	
Egocentric Spatial ZUI [RJ+13]	x	x	x	
Code Space [BD11]	x	x	x	
Touch Projector [BB+10]	x	?	?	
Context-Aware Applications [SAW94]	x	?	?	?
Sentient Computing System [AC+01; HH+02]	x	x	x	x
DiffDisplays	x	x		
MultiView Train Board	x	x	x	x
MultiView Video Player		x	x	
SpiderEyes		x	x	

Table 3.4: Interaction - Range

Value legend: x - System was described or demonstrated in use within this distance range; ? - System was not explicitly demonstrated or described as used within this distance range but its use within this range was implied elsewhere in the description or by the characteristics of its sensors or deployment site.

3.6.1 Intimate Range (<0.6 m)

In most cases, the intimate range interaction involved some form of contact-based input, whether it was with a touch enabled display or with a keyboard or a similar input device. This was true for 23 out of the 26 systems using this range. The non-contact systems were *Opo* (where distances down to 0.25 m were collected), *MultiView Train Board* (where it was possible for people to stand directly

under the display), and *Psychic Space/Maze* (where the entirety of the room was instrumented[11]). That is not to say that no other interactions took place in this range in systems that did use contact-based input. A number of systems made use of various device-based gestures, e.g. tilting a mobile phone in a specific way to trigger an action. However, in all cases these techniques involved some form of manipulation of a mobile device.

3.6.2 Personal Range (<1.5 m), Social Range (<5 m) and Public Range (>5 m)

All systems but one used Personal Range range for interaction. A large number of systems (27) made use of the Social Range, but only six systems actively used the Public Range. There does not seem to be specific patterns in terms of why some of these ranges were used while others were not, aside from two general trends. The first one is the focus of the systems, where the systems only implemented interactions at ranges that were necessary to fulfil the goal of the system, or that made sense in terms of the systems design goals. The second reason is a more prosaic one, where especially at the larger ranges the limitations were generally imposed by sensor constraints. This is clearly visible especially in early systems. We speculate that it is likely that if Ju et al. [JLK08] had access to a Kinect sensor, they would most likely use it for their design of the *Range Whiteboard* to extend their interaction range beyond 1.5 m. Other systems were sensor limited even with the use of Kinects, for example *ProximityTable* used a ceiling mounted Kinect sensor, which limited the interactive area due to the relatively low ceiling in the interactive space. This was compounded by inclusion of a tabletop display in the interactive space. As will be demonstrated later on in this section, two systems (out of five) that made use of the Public Range did so more as a side-effect of their physical design than necessarily because this was intended by the designers of the systems.

3.6.3 Systems Covering All Ranges

Five systems utilised all four ranges. Three of those systems, *Authenticated Badge*, *Context-Aware Applications*, and *Sentient Computing System*, were large scale systems, sometimes spanning multiple buildings. The *MultiView Train Board* was designed with long distance (albeit passive) interaction in mind, with the interaction in the Public Range being rather limited. However, with *Psychic Space/Maze*, the use of the Public Range was purely due to the spatial coverage rather than explicit design intention. The tracking area of the system was 4.9×7.3 m. Both *MultiView Train Board* and *Psychic Space/Maze* are also non-contact systems, where interaction within the Intimate Range only involves body movement.

3.6.4 Systems Covering All Ranges Except Public

Systems making use of the Intimate, Personal and Social Ranges were the most common with fifteen systems in this cluster. To give the descriptions more structure, the systems were divided into one of four themes - smart spaces, wall displays, tabletops and multi-display environments, and other systems.

In the smart spaces grouping are *Easy Living*, *GroupTogether*, *Code Space*, *Proxemic Controls*, and *Proxemic Media Player*. In all cases, most of the interaction techniques were performed at the Intimate and Personal Ranges and generally the Social Range use was due to the spatial design of the interactive space rather than being specifically targeted by system designers. Frequently, the same techniques were used in both the Personal and Social Ranges.

The next group of systems all use an interactive wall display as one of the defining characteristics. Both *Hello.Wall* and *Public Ambient Displays* use spatial relationships at the Personal and Social

[11] However, it could be argued that since with *Psychic Space/Maze* the person interacting was walking on the pressure sensitive tiles, the tiles are a contact-based input.

Ranges to change between interaction zones that define interaction modes. These systems, together with *Proxemic Peddler* which is also in this group, use direct touch interaction with the wall display as the main interaction in the Intimate Range. *Egocentric ZUI*, the last system in the group, uses direct touch interaction with a tablet rather than with the wall display in the Intimate Range. Unlike the other systems, this is the only interaction technique used within the Intimate Range.

The third group of systems all have tabletop display as one of their interactive elements. All three systems, *E-conic*, *Medusa*, and *LightSpace* use touch in the Intimate Range. In *E-conic* spatial interaction was mainly for perspective correction throughout all three ranges. *LightSpace* also used only spatial interaction beyond the Intimate Range. The same was the case with *Medusa*, but in addition to direct touch, the system also explored hover interaction in the Intimate Range.

The last group of systems is more eclectic. *Opo* is strictly a data collection system, so any interaction is very passive. The only reason why the system is also classified as using the Social Range is because the maximum sensed distance was 2 m. For completeness, the minimum distance was 0.25 m, which is in the Intimate Range. *Vision Kiosk*, a public kiosk system, uses direct touch in the Intimate Range, and the description of the system implies the use of the Personal and Social Ranges, although this could not be verified. *Touch Projector* also uses direct touch in the Intimate Zone, beyond which only spatial information is used. The distance range for remote interaction was constrained by the video capabilities of the system and based on the experimental setup, the range is approximately from 0.6 m to 4 m, which covers the Personal and Social Ranges.

3.6.5 Systems Using Personal and Social Ranges

There were six systems using only a combination of Personal and Social Range — *Puppeteer Display*, *SpiderEyes*, *Proxemic InfoVis*, *Is Anyone Looking?*, *MultiView Video Player*, and *Shadow Reaching*. These systems were generally very similar to the systems described in the section just preceding this one except interactions with these systems do not involve either a touch enabled mobile device or a touch enabled wall display or tabletop. All of the systems in this group fit into the large display theme.

3.6.6 Systems Using Intimate and Personal Ranges

The five systems using only Intimate and Personal Ranges for interaction can be divided into two sub-groups — personal systems and sensor-limited systems. *DiffDisplays*, *Lean & Zoom*, and *Active Hydra* are all systems, which were primarily designed to be used by one person at a time within their office setup. *Active Hydra* is used for remote communication but each of the Hydras is expected to be used by a single individual. Therefore, it does not come as a great surprise that the systems do not explore Ranges beyond the Personal Range as desk-based interaction spaces generally do not extend beyond 1.5 m from the user.

The sensor-limited group contains *ProximityTable* and *Range Whiteboard*. In both cases, it is clear from the description of the systems that if the authors had access to sensor technology that would allow them to track spatial position beyond the tracked area they had available, the systems would have made use of this additional information. In the case of *ProximityTable*, the tracked area was only 120×150 cm, which included the tabletop display as well. *Range Whiteboard* was limited by the distance sensors on the board to a maximum detected distance of 150 cm from the board.

3.6.7 Others

The last two systems do not fit well into any of the other clusters. *Screenfinity* falls somewhere between the other clusters as it uses the Personal, Social and Public Ranges. However, it would fit most closely among the systems using only Personal and Social Ranges as the only reason why *Screenfinity* is listed as using the Public Range is due to the display itself being 5 m wide and the

Name	Cardinality											
	Actor				Object				Environment			
Opo [HK+14]		1*										
Puppeteer Display [BB+14]			M	M*								
GroupTogether [MHG12]	1	1*	M	M*	1	1*	M		1	1*		
Is Anyone Looking? [BL+14]		1*			1							
ProximityTable [Aln15; HR+14]	1	1*	M	M*	1							
LightSpace [WB10]	1	1*	M	M*	1		M					
Proxemic Media Player [BMG10]	1	1*	M		1	1*	M		1			
Hello.Wall [SP+03; PR+03]	1	1*			1	1*						
Public Ambient Displays [VB04]	1	1*			1							
Proxemic InfoVis [JS+13]	1				1							
Medusa [AG+11]	1	1*			1							
Proxemic Peddler [WBG12; Wan12]			M									
Mobile Sensing Techniques [HP+00]	1				1							
Vision Kiosk [CA00]	1		M		1							
Screenfinity [SMB13]	1	1*	M		1							
E-conic [NS+07]	1	1*	M	M*			M	M*				
EasyLiving [BM+00; KH+00; BS09]	1	1*			1	1*	M					
Active Hydra [Gre99; KG99]	1		M		1	1*	M		1			
Range Whiteboard [JLK08]	1		M[1]		1							
Authenticated Badge [WH92; WH+92]	1	1*	M	M*	1	1*			1	1*	M	M*
Lean & Zoom [HD08]	1				1							
Shadow Reaching [STB07]	1	1*			1							
Psychic Space/Maze [Kru77; KGH85]	1				1				1			
Proxemic Controls [Led14; LGB15; GL13]	1	1*			1	1*	M		1			
Egocentric Spatial ZUI [RJ+13]	1						M					
Code Space [BD11]	1	1*	M	M*	1		M					
Touch Projector [BB+10]	1						M					
Context-Aware Applications [SAW94]		Any				Any				Any		
Sentient Computing System [AC+01; HH+02]		Any				Any				Any		
DiffDisplays	1				1	1*						
MultiView Train Board	1	1*	M[1]	M*	1							
MultiView Video Player	1		M		1							
SpiderEyes	1	1*	M	M*	1							

[1] Only one person in the group can actively interact

Table 3.5: Interaction - Cardinality

sensors being positioned in such a way that the tracked area was up to 14 m in length. Aside from this, the system does not actively use the Public Range for any specific interaction techniques.

Mobile Sensing Techniques stands out as it only uses the Intimate Range. This is due to the system being focused on direct interaction with a mobile phone and because the distance sensors detected values only up to 25 cm.

3.7 Interaction - Cardinality

While there are potentially 64 combinations of values for cardinality, in practice there are two defining characteristics for each entity type. The first one describes whether the system can accommodate groups of the same entity type. The second one denotes whether the system supports parallel interaction by multiple entities of the same type. We can simplify the values to this combination because in order for an entity type to have any cardinality value, at least one entity of that type

needs to be supported by the system. The parallel and group/connected values essentially act as multipliers that capture the systems's capabilities. Therefore the process for arriving at the expected set of cardinality values for a particular entity type is as follows:

1. If at least one interactive entity of this type is supported by the system, add `1` to the set of values.
2. If groups of this entity type are supported, add `M` to the set of values.
3. If multiple entities of this type can interact in parallel, add * variants of any already existing values to the set.

In some cases, the result will not quite match the values in Table 3.5, which shows the values for all the analysed systems. This helps to identify unusual and potentially interesting systems. All of the systems that diverge from the expected values are discussed in more detail in this section. However, generally speaking, the reasons for the divergence from expected values tend to be either due to spatial constraints, potentially incomplete description of the system, or a specific (potentially unique) characteristic of the system. The results of cardinality analysis are structured according to entity types.

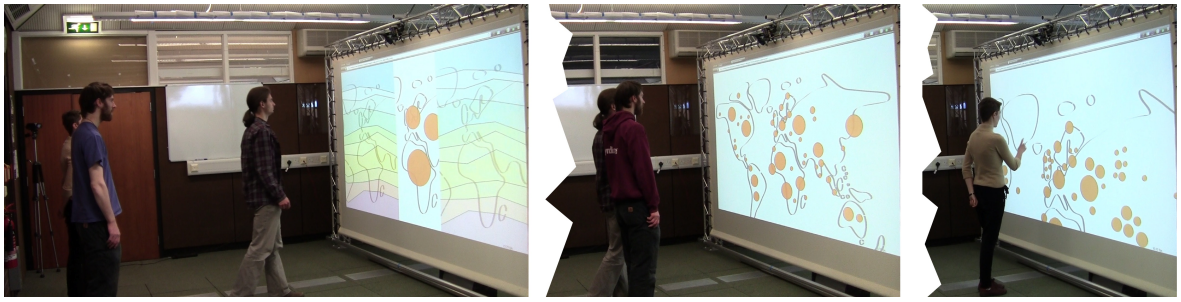


Figure 3.3: An illustration of cardinality using *SpiderEyes*. The subfigure on the left shows two persons interacting in parallel (`1*`), while the subfigure in the middle shows two people interacting as a group (`M`). The right image shows a single person interacting (`1`).

3.7.1 Actors

All systems have Actors as interactive entities. If a system did not support Actor interaction, it would indicate a problem with the analysis. This is because Actors are defined as the entities driving interactions with the system and if there were no entities capable of active interaction, there would be no possibility of interactions taking place.

Twenty-one systems support parallel interaction by Actors in some form. With *Opo*, this was a simple case of multiple people being instrumented for data collection. *Is Anyone Looking?* concentrates on parallel use by two individuals as the interaction techniques centre on a primary user of the system and a second person, who is likely a shoulder-surfer, attacking the primary user's privacy. A large group of systems, namely *ProximityTable*, *LightSpace*, *Hello.Wall*, *Public Ambient Displays*, *Medusa*, *Screenfinity*, *E-conic*, *EasyLiving*, *Shadow Reaching*, and *Proxemic Controls* support simultaneous parallel interaction by multiple individuals, although with a number of the systems some of the resources are physically shared (e.g. people interact with different parts of a large display).

Eighteen systems allow for collaborative group interactions. In nine of the eighteen systems, it appears only a single group is supported, even in cases where the system also supports parallel interaction. In case of *Active Hydra*, this is due to a deployment limitation as only two *Active Hydras*

were in use. For *Range Whiteboard*,*Vision Kiosk*, and *MultiView Video Player*, this was due to an assumption in the system design where if multiple persons were present, they were treated as a single group, regardless of their actual connections.

Puppeteer Display and *Proxemic Peddler* are rather unique systems, which only register group values for cardinality. This is because in both systems, the displays are Actors in the interactions and so any interaction is by definition a group interaction because both the display and the person simultaneously influence each other. In the case of *Puppeteer Display*, the display actually supports interaction with multiple people simultaneously, which creates an interesting case where the display is simultaneously part of multiple otherwise interactionally independent groups.

Of the eighteen systems, ten fully supported interaction by groups, even multiple ones. With the exception of *Puppeteer Display*, which was described earlier, all of the systems in this group support all four possible interaction cardinalities. Three of the systems are large-scale systems that allow for virtually any number and combination of Actors (*Authenticated Badge*, *Context-Aware Applications*, and *Sentient Computing System*). *SpiderEyes* is not a large-scale system, but it supports parallel use and since there is no constraint on how many groups may be formed, it is possible to have any combination of Actors. However, the system was designed with a practical limit of three to four simultaneous users due to spatial constraints. *LightSpace* and *ProximityTable* theoretically support any combination of Actors but the capability was not well demonstrated in the available materials. *MultiView Train Board* supports any combination of Actors but this is a side-effect of the system design, which allows for simultaneous viewing of multiple sets of information without the need for explicit tracking of all present people.

With *Code Space*, the group values depend on scenario interpretation. On one hand, all users present in the room form a group as they are all working on the same task (source code review). Moreover, some techniques require collaboration between two people to be successfully executed[12]. On the other hand, most of the interaction techniques only require actions by a single user. The reverse is true for *GroupTogether*. The system concentrates on pair and group interaction techniques, but it is also possible for an Actor to be a singleton or a bystander and thus the `1` and `1*` values are also possible.

To complete this section, seven systems, namely *Proxemic InfoVis*, *Mobile Sensing Techniques*, *Lean & Zoom*, *Psychic Space/Maze*, *Egocentric ZUI*, *Touch Projector*, and *DiffDisplays*, were demonstrated only with a single Actor entity.

3.7.2 Objects

Object cardinality is defined per Actor. While it may be possible for a system to have, for example, multiple sets of connected Objects, unless they can be simultaneously used by an Actor, the system will not be classed as `M*` for Object cardinality. Unlike with Actors, it is entirely possible for a system to not have any Objects within its interactive scenarios. The three systems that fall into this category are *Opo* (which only captures interactions between people), *Puppeteer Display* and *Proxemic Peddler*, where the interactive display is also an Actor due to its ability to exhibit control over the interaction.

Fifteen systems only use a single object in their interaction scenarios. In most cases, this was a large or shared display of some kind. However, for *Medusa* and *ProximityTable* it is a tabletop display. For *Lean & Zoom* it is a laptop display, while in *Vision Kiosk* it is a public kiosk. Lastly, *Mobile Sensing Techniques* uses a single augmented mobile phone.

Three systems only allow use of sets of interconnected Objects. In *E-conic*, all the displays in the multi-display environment are connected to form a canvas that allows seamless viewing of information. With *Egocentric ZUI*, the large wall display and the tablet used as a lens are connected

[12]Since multiple pairs may be performing those techniques simultaneously, the `M*` cardinality is also possible.

and cannot be used independently. *Touch Projector* directly translates touch input from a mobile phone onto a remote display, which requires them to be connected.

Another three systems allow interaction with a single Object, or with multiple independent Objects in parallel. In *Hello.Wall*, a person can interact with the large display or with the ViewPort. *DiffDisplays* can be used with either a single display or with multiple displays. *Authenticated Badge* allows users to interact with telephones and workstations simultaneously and they are not directly connected.

Two systems support interaction with either a single Object, or a set of connected Objects. *LightSpace* allows Actors to connect Objects by touching them, creating a connected set. *Code Space* allows people to use their mobile devices independently, or to perform sharing techniques, which connect pairs of devices.

Five systems allow interaction with a single Object, a set of connected Objects or multiple independent Objects. *GroupTogether*, *Proxemic Media Player*, *EasyLiving*, *Proxemic Controls*, and *Active Hydra*. In *Active Hydra*, the surrogates can be used independently one at a time, or simultaneously. However, they can also be connected together to augment the Active Hydra unit. The other three systems all allow a mobile device and a large display to be used simultaneously as independent Objects, or connected together as part of an interaction technique. Each Object can also be used by itself.

Lastly, two systems have the potential to let Actors to use any combination of Objects. These are *Context-Aware Applications* and *Sentient Computing System*, both of which are large infrastructure-based projects. While the `M*` cardinality has not been demonstrated, the systems clearly have the potential to enable any combination of Object interactions.

3.7.3 Environments

Most systems do not actually use Environments for interaction. The eight systems that do essentially fall into three groups. The first group consists of large scale infrastructure-based systems, namely *Context-Aware Applications*, *Sentient Computing Systems*, and *Authenticated Badge*. In all cases, since the systems' infrastructure spans multiple rooms, even buildings, multiple environments are present. Since the system can connect entities present in different Environments, this creates an implicit connection between the Environments.

Four systems use relationships with a single Environment - *Proxemic Media Player*, *Active Hydra*, *Psychic Space/Maze*, and *Proxemic Controls*. It could be argued that the 23 systems that are not classified as using an Environment are using an Environment implicitly. However, no spatial relationships with Environments are used by the systems, and in most cases the concept of an Environment (or a similar concept) is not found in the descriptions of the systems. In all four systems that use a single Environment, the entity corresponds to a room.

The last system that uses Environments is *GroupTogether*. The system is somewhat unique, in that the interactive Environment generally corresponds to a group of people rather than the room. In addition, since multiple groups can be formed but they are not otherwise connected, the system supports both a single Environment and parallel independent Environments.

3.8 Interaction - Mode

The analysis of interaction modes reveals two main outcomes. Firstly, most systems used similar modes in similar ways. However, the way modes are matched between entities showed several interesting results. To begin with, the different modes as they were used by the systems will be described. Spatial mode was used by every single system by definition and the details of its use are captured in the spatial relationship parts of the analysis. Visual mode was utilised by the vast

	Mode											
Name	Actor					Object					Environment	
Opo [HK+14]	Sp											
Puppeteer Display [BB+14]	Sp	Vis										
GroupTogether [MHG12]	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym		Sp	
Is Anyone Looking? [BL+14]	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym			
ProximityTable [Aln15; HR+14]	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym			
LightSpace [WB10]	Sp	Vis	Int			Sp	Vis	Int				
Proxemic Media Player [BMG10]	Sp	Vis	Int	Sym	Ac	Sp	Vis	Int	Sym	Ac	Sp	
Hello.Wall [SP+03; PR+03]	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym			
Public Ambient Displays [VB04]	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym			
Proxemic InfoVis [JS+13]	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym			
Medusa [AG+11]	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym			
Proxemic Peddler [WBG12; Wan12]	Sp	Vis	Int	Sym								
Mobile Sensing Techniques [HP+00]	Sp	Vis	Int	Sym	Ac	Sp	Vis	Int	Sym	Ac		
Vision Kiosk [CA00]	Sp	Vis	Int	Sym	Ac	Sp	Vis	Int	Sym	Ac		
Screenfinity [SMB13]	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym			
E-conic [NS+07]	Sp	Vis	Int			Sp	Vis	Int				
EasyLiving [BM+00; KH+00; BS09]	Sp	Vis	Int	Sym	Ac	Sp	Vis	Int	Sym	Ac		
Active Hydra [Gre99; KG99]	Sp	Vis	Int		Ac	Sp	Vis	Int		Ac	Sp	
Range Whiteboard [JLK08]	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym			
Authenticated Badge [WH92; WH+92]	Sp	Vis	Int		Ac	Sp	Vis	Int		Ac	Sp	
Lean & Zoom [HD08]	Sp	Vis				Sp	Vis					
Shadow Reaching [STB07]	Sp	Vis	Int			Sp	Vis	Int				
Psychic Space/Maze [Kru77; KGH85]	Sp	Vis					Vis				Sp	
Proxemic Controls [Led14; LGB15; GL13]		Vis	Int	Sym		Sp	Vis	Int	Sym		Sp	
Egocentric Sp ZUI [RJ+13]	Sp	Vis		Sym	Ac	Sp	Vis		Sym	Ac		
Code Space [BD11]	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym			
Touch Projector [BB+10]	Sp	Vis	Int			Sp	Vis	Int				
Context-Aware Applications [SAW94]			Any					Any				Any
Sentient Computing System [AC+01; HH+02]			Any					Any				Any
DiffDisplays	Sp	Vis	Int	Sym		Sp	Vis	Int	Sym			
MultiView Train Board	Sp	Vis		Sym		Sp	Vis		Sym			
MultiView Video Player	Sp	Vis		Sym	Ac	Sp	Vis		Sym	Ac		
SpiderEyes	Sp	Vis		Sym		Sp	Vis		Sym			

Table 3.6: Interaction - Mode

Value legend: Sp - Spatial, Vis - Visual, Int - Intent, Sym - Symbolic, Ac - Acoustic

majority of systems and in all cases was due to the system incorporating some sort of a display showing visual output. Intent mode was utilised by systems offering either direct touch interaction, or some form of remote pointing or gaze interaction. Systems that made use of the Symbolic mode most commonly did so by showing some form of textual or otherwise symbolic information on displays. Other systems allowed gestural input. One system used speech input and output. Ten systems made use of the Acoustic mode, using non-linguistic sounds. No systems made use of any of the other modes mentioned in the IRE model.

Modes are always used between two entities, an emitter and an interpreter. There may be multiple interpreters but in that case, the communication can be divided into sets of communicating pairs where one entity of each pair is the emitter. As can be seen in Table 3.6, which shows all the mode values for each system, with most systems any modal interaction happens between Actors and Objects, in some combination (including Actor-Actor and Object-Object). However, some systems showed somewhat unusual matching of modes between entities. Three systems used modal interactions only between Actors. In two of these systems, *Puppeteer Display* and *Proxemic Peddler*, displays are Actors too. In the third system, *Opo*, the interaction is passively tracked.

The second interesting set of systems are those that use Environment entities, there are eight of them. Two systems, *Context-Aware Applications* and *Sentient Computing System* can potentially

use any mode with an Environment entity as they are very flexible and potentially open ended systems. However, from the system descriptions it was not clear which of the modes were utilised. Of the remaining six systems, *GroupTogether* paired the Environment to both Actors and Objects due to the Environment being defined as a group of interacting people.[13] *Proxemic Media Player* and *Authenticated Badge* used two pairs for the Spatial mode with Environments - Actor-Object and Actor-Environment. *Active Hydra* and *Proxemic Controls* only used the Object-Environment pair, again with the Spatial mode. *Proxemic Controls* is worth pointing out as it is the only system that did not use the Spatial mode for Actors at all. Lastly, *Psychic Space/Maze* uses only the Actor-Environment pair for Spatial mode pairing with Environment entities.

The main outcome of analysing modes is that most systems that used Environment entities only used them with Spatial mode. Most other systems only used modes in Actor-Actor, Actor-Object or Object-Object pairs. Another notable observation is that for some of the systems some of the utilised modes are actually side-effects of the applications rather than primary intent of the system designers (e.g. *MultiView Video Player* uses the Acoustic mode but this is only because the example scenario is video playback, the system does not need sound for its primary/defining interaction techniques.

Similar situations can be found in most other systems, with a few notable exceptions. *Proxemic Media Player* uses sounds to indicate that the system is aware of a potential user and that it is transitioning into active mode. Other systems tend to use the Acoustic mode as a side-effect of an application, e.g. music playback. *MultiView Video Player*, *SpiderEyes*, *E-conic*, *Screenfinity*, *Proxemic InfoVis* actively modify the size of textual and symbolic information to keep it readable. Other systems mostly use textual and other symbolic information to demonstrate other techniques.

3.9 Interaction - Intentionality

Intentionality of an action can be accurately determined only on the side of the emitter of the action. However, in practice this is generally not practically possible. Some systems do not seem to take intentionality of actions into account at all. In cases where the intentionality is considered, most systems attempt to infer the intentionality from the sensor values or input methods, or the systems simply assume that if certain conditions are met, the action is of specific intentionality. With inference-based methods, there is always a certain probability of erroneous interpretation of the data. Moreover, any sensing or input method has edge cases, which will produce erroneous input values.

These complications, coupled with limited descriptions of systems, lead to this analysis being inherently limited in its accuracy due to the relatively low reliability of the data available. In order to increase the signal-to-noise ratio of the analysis, an overall threshold for the expected robustness of intentionality values was determined. Any output by a computer system is with almost complete certainty explicit. Input using standard input devices such as keyboards, mouse or direct touch was generally deemed sufficiently resilient to misinterpretation. Similarly, gesture based systems were also assumed to be robustly implemented, although this could not be directly confirmed. Any exceptions to this are explicitly described in this section. Interactions using spatial information were judged individually, with focus on whether the designers of the system accounted for potential spurious inputs or if they clearly described how the system determined the intentionality of the relevant actions. Where no such information was available or where there was an indication that the actions were interpreted in a manner likely to produce erroneous interpretations, this is marked accordingly by using the *explicit+* and *implicit+* intentionality types.

Interestingly, authors of several systems note that their system design leads to misinterpretations of actions. Authors of *Proxemic InfoVis* acknowledge the need for a locking mechanism to deal with interaction problems due to misinterpreting user actions. The authors of *Vision Kiosk* also share

[13] The shared display was an Object, but it could form a part of the group.

Name	Intentionality											
	Actor				Object				Environment			
Opo [HK+14]		E+										
Puppeteer Display [BB+14]	E	E+		I+								
GroupTogether [MHG12]	E			I+	E	I						A
Is Anyone Looking? [BL+14]	E	E+			E							
ProximityTable [Aln15; HR+14]	E			I+	E							
LightSpace [WB10]	E				E							
Proxemic Media Player [BMG10]	E	E+		I+	E		I+					A
Hello.Wall [SP+03; PR+03]	E			I+	E							
Public Ambient Displays [VB04]	E	E+		I+	E							
Proxemic InfoVis [JS+13]		E+			E							
Medusa [AG+11]	E			I+	E							
Proxemic Peddler [WBG12; Wan12]	E	E+										
Mobile Sensing Techniques [HP+00]	E	E+			E		I+					
Vision Kiosk [CA00]	E	E+			E							
Screenfinity [SMB13]		E+			E							
E-conic [NS+07]	E		I		E							
EasyLiving [BM+00; KH+00; BS09]	E			I+	E		I+					
Active Hydra [Gre99; KG99]	E	E+			E							A
Range Whiteboard [JLK08]	E	E+			E							
Authenticated Badge [WH92; WH+92]	E			I+	E							A
Lean & Zoom [HD08]	E	E+			E							
Shadow Reaching [STB07]		E+			E							
Psychic Space/Maze [Kru77; KGH85]		E+			E							A
Proxemic Controls [Led14; LGB15; GL13]	E				E		I+					A
Egocentric Spatial ZUI [RJ+13]		E+			E							
Code Space [BD11]	E		I		E							
Touch Projector [BB+10]	E				E							
Context-Aware Applications [SAW94]			Any				Any				Any	
Sentient Computing System [AC+01; HH+02]			Any				Any				Any	
DiffDisplays	E		I		E							
MultiView Train Board		E+			E			A				
MultiView Video Player	E				E			A				
SpiderEyes		E+		I+	E							

Table 3.7: Interaction - Intentionality

Value legend: E - Explicit, E+ - Explicit (with possible misclassifications), I - Implicit, I+ - Implicit (with possible misclassifications), A - Ambient

examples of misinterpretations, where for example in one deployment the system essentially failed to establish who the avatar should interact with due to too many people passing the display within the distance used to assume intention to interact. While such observations do not provide a solution, it is clear that at least some authors and system designers are aware of these issues.

Table 3.7 shows the intentionality for all systems. Two systems, *Context-Aware Applications* and *Sentient Computing System*, are covered separately from the rest of the systems due to their nature. While only some of their applications were described by the authors, the system descriptions make it clear that the systems are likely to cover virtually every possible intentionality value. The results for these two systems are not counted to the system totals when presenting intentionality values. In order to provide structure to the presentation of results for the remaining systems, they are presented based on the entity type performing the analysed action.

3.9.1 Actors

Actor actions show the greatest variety. All systems consider at least some Actor actions as *explicit*. In 23 of the systems, these actions are likely to be interpreted correctly. This is mostly due to the systems using standard inputs (direct touch, keyboard, mouse, mid-air gestures, physical buttons). The exceptions to this are *Puppeteer Display*, *Proxemic Peddler*, *Hello.Wall*, *Active Hydra*, *GroupTogether*, and *MultiView Video Player*. With *Puppeteer Display* and *Proxemic Peddler* it is due to the displays being classed as Objects, so their explicit outputs are classed here (*Proxemic Peddler* otherwise uses direct touch interaction as well). *Hello.Wall* and *Active Hydra* use inputs that are unlikely to be misinterpreted even though they are not considered to be standard inputs (very short range RFID tags for *Hello.Wall* and manipulation of physical figurines for *Active Hydra*). *GroupTogether* is singled out because its authors were one of the very few that considered (and more importantly tested) how likely it was for the technique triggers to be activated accidentally. Lastly, *MultiView Video Player* uses seating location to determine whether subtitles are visible, it is unlikely that a person would sit down accidentally.

While in large portion of the systems, some actions were classified as safely *explicit*, this was generally due to the analysis threshold described in the beginning of this section rather than due to the authors considering the likelihood of false positives. However, even with the threshold in place, 19 systems had at least one set of Actor actions, which were classified as likely to be prone to misinterpretations (classed as *explicit+*). The reason for this was uniformly because the system designers did not demonstrate that any consideration to the accuracy and reliability of the sensors or the likelihood of actions being incorrectly interpreted. Additionally, most of the affected actions are spatial actions used to trigger techniques (changes in distance, orientation, or position).

Only three systems use actions *implicitly* in a way unlikely to lead to misinterpretation. *E-conic* uses people's point of view to correct the perspective of any of the application windows. It is highly unlikely that the tracked person would intentionally want to view the windows in a skewed manner. *DiffDisplays* tracks a person's orientation to tracked displays, in order to establish whether they are looking at a display or not. The system was designed in a way that the person is highly unlikely to be able to see any detail on the display if they are judged as not looking at the display (although the display may still be in their field of vision). *Code Space* is somewhat similar, in that in one of the techniques, a presented specific UI is shown on a shared display when the presenter is facing the display.

Ten systems use actions *implicitly* but with a risk of the actions being misinterpreted. This is exclusively due to the system designers' making assumptions about the triggers for their systems, e.g. assuming that a certain distance between individuals implies that they want to form a group (*GroupTogether*, *ProximityTable* and *SpiderEyes*). Heuristic approaches to technique triggers and action interpretations are likely to produce good enough results for an evaluation of a research prototype, but they tend to lead to exactly the kinds of misinterpretations that can make systems problematic to use.

3.9.2 Objects

All systems, with the exception of *Puppeteer Display* and *Proxemic Peddler*, which have already been covered, include at least one display classed as an Object. The display output is robustly *explicit* within this analysis. Five systems use the changes in spatial state of a device as an *implicit* action. Four of the five systems are classed as *implicit+* as their use of the action is likely prone to false positives. The only system classified as having a robust *implicit* action detection is *GroupTogether* as the authors actively considered how prone their system was to accidental triggering of the interaction techniques.

Two prototypes developed for this thesis demonstrated *ambient* actions with Objects by exploiting the physical properties of the displays used in the prototypes, specifically the ability to multiplex simultaneous views from a single display. The spatial context of the display in *MultiView Video Player* determined whether an Actor could see subtitles or just the video track. No action was performed by the display. *MultiView Train Board* utilised the same properties to present an ambient static view, which was visible from large distances, in addition to the interactive view, which was visible from shorter distances.

3.9.3 Environments

There are eight systems that use an Environment entity. The two large-scale systems, *Context-Aware Applications* and *Sentient Computing System* can potentially use any intentionality with their Environments. However, like the other six systems, only ambient intentionality has been demonstrated. In all cases, the Environments are only used to derive contextual information rather than any actual actions being performed by the Environments.

3.9.4 Ambient Displays

In addition to the analysis discussion above, one more topic is worth mentioning. Three of the analysed systems, *Code Space*, *Hello.Wall* and *Public Ambient Displays*, claim to have some form of "ambient" mode. While this term is quite commonly used, it can lead readers to misinterpret the behaviour of the display, at least in the context of this analysis. Within the IRE model, *ambient* actions are those where only contextual information derived from the entity's state is used and no action takes place. However, with most of the systems with "ambient" mode this is not the case, at least according to the system descriptions. The displays still actively change the content on the displays, only showing non-personalised or non-interactive content. This constitutes an action and thus by definition the display is not *ambient* from the IRE model point of view. In practice a term such as "non-interactive" may more accurately describe the display mode.

This is in contrast to the three "ambient" systems with *MultiView Video Player* and *MultiView Train Board*, which do not perform any actions in order to provide an ambient view. Arguably, *MultiView Video Player* does not fully adhere to this as video playback could still be interpreted as an action by the display (the contents of the display are changing). *MultiView Train Board* does not have any such limitation.

3.10 Interaction - Intensity

Intensity has proven a very complex measure to analyse. This is due to two main reasons. Firstly, the descriptions of systems and interactive techniques generally do not provide sufficient information to accurately establish the likely impact of an action. For example a visual change to content on a display will have varying levels of impact depending on the physical properties of the display and the positions and orientations of all entities able to visually perceive the display. System descriptions do not tend to go into this level of detail. Therefore, the values shown in Table 3.8 show estimates by the author based on the information available (textual descriptions, images, video footage).

The second issue is that even if the descriptions provided sufficient detail, the same action can result in different intensity of its effects at different points in time depending on other circumstances. For example, the vibration of a mobile phone can prove very subtle or even unnoticeable when the attention of the person holding is directed elsewhere. However, it is entirely possible for the same vibration under the same circumstances to be disruptive if the person holding the phone is startled by it.

Name	Intensity									Environment
	Actor					Object				
Opo [HK+14]	U									
Puppeteer Display [BB+14]		S	N				S	N		
GroupTogether [MHG12]			N					N		
Is Anyone Looking? [BL+14]		S	N				S	N	I	
ProximityTable [Aln15; HR+14]			N					N		
LightSpace [WB10]			N					N		
Proxemic Media Player [BMG10]			N	I	D		S	N		
Hello.Wall [SP+03; PR+03]			N				S	N		
Public Ambient Displays [VB04]		S	N			U	S	N		
Proxemic InfoVis [JS+13]			N				S	N		
Medusa [AG+11]			N				S	N		
Proxemic Peddler [WBG12; Wan12]		S	N							
Mobile Sensing Techniques [HP+00]		S	N				S	N		
Vision Kiosk [CA00]			N				S	N		
Screenfinity [SMB13]			N				S	N		
E-conic [NS+07]			N					N		
EasyLiving [BM+00; KH+00; BS09]			N			U	S	N		
Active Hydra [Gre99; KG99]		S	N				S	N		
Range Whiteboard [JLK08]			N				S	N		
Authenticated Badge [WH92; WH+92]		S	N				Any			
Lean & Zoom [HD08]			N					N		
Shadow Reaching [STB07]			N					N		
Psychic Space/Maze [Kru77; KGH85]			N					N		
Proxemic Controls [Led14; LGB15; GL13]			N				S	N		
Egocentric Spatial ZUI [RJ+13]			N					N		
Code Space [BD11]			N					N		
Touch Projector [BB+10]			N					N		
Context-Aware Applications [SAW94]			Any					Any		Any
Sentient Computing System [AC+01; HH+02]			Any					Any		Any
DiffDisplays			N			U	S	N		
MultiView Train Board			N			U		N		
MultiView Video Player			N			U	S	N		
SpiderEyes			N			U	S	N		

Table 3.8: Interaction - Intensity

Value legend: U - Unnoticeable, S - Subtle, N - Neutral, I - Intrusive, D - Disruptive

In order to make the analysis result more descriptive given the limitations described above, the results in Table 3.8 show the most frequent estimated intensity values rather than all possible values as it is theoretically possible for any action to fall anywhere on the intensity spectrum given the right set of circumstances.

With these limitations in mind, let us consider the results. The vast majority of systems seem to fall into the Neutral range on the intensity spectrum. A significant portion of the systems track and emit actions that are likely to be subtle. For example, *Public Ambient Displays* uses changes in the head orientation of a passer by (Actor), which can be quite subtle, to initiate transition to a more interactive mode. An example for Objects from the same system could be that the width of a person's interactive space on the large display changes based on the body orientation of the person.

In some cases, the systems perform actions which cannot be noticed by people interacting with the system, rendering them Unnoticeable. Take *DiffDisplays* for example. The techniques for tracking

visual changes on unattended displays run while the Actor is not looking towards the display. This means that the person will only even see the last few moments of the techniques and only after they look back at the display. On the opposite end of the spectrum, in the two systems that likely contain actions that are either Intrusive or Disruptive, the actions are due to an interactional conflict between two Actors. In *Proxemic Media Player*, one person's movie playback is interrupted when another person takes over the shared display. In *Is Anyone Looking?*, this is even more explicit as the disruption occurs when the system tries to protect sensitive content on the display from a potential shoulder-surfing attacker. Most of the techniques would cause some level of disruption to the primary user.

Unfortunately the analysis of action intensity is somewhat incomplete due to generally insufficient level of detail within the information available. Even with additional materials in the form of images and sometimes video footage, in a large number of cases it was impossible to determine the likely impact of a significant portion of the interactive actions. Where it was not possible to establish a value, a normal/neutral value is assumed. This situation illustrates an issue regarding insufficient detail in system descriptions, discussed in more detail in Section 3.11.

3.11 Discussion, Trends and Opportunities

The analysis revealed a number of trends, as well as unique properties of existing systems, both of which can be utilised to identify opportunities for future research explorations. In this section, the main trends and outcomes are summarised together with suggestions of how to exploit them.

Proxemics A number of system authors identified the notion of proxemics as a guiding idea in the design of their systems. Proxemics, as defined in Edward Hall's book *The Hidden Dimension*, is "the interrelated observations and theories of man's use of space as a specialized elaboration of culture" [Hal66]. Hall's primary focus was linguistic and cultural. However, his focus was also on people, even though he devotes a significant portion of the book to animal behaviours. With that in mind, it was somewhat disappointing to find that the vast majority of systems did not make any use of spatial relationships between people. In fact, the most common relationships were those between people and interactive objects (Actor-Object, Object-Actor). Actor-Actor relationships were no more commonly used than Object-Object relationships (especially one systems where displays were classified as Actors are accounted for).

This trend can likely be at least partially explained through sensor limitations, as systems allowing for accurate tracking of people within large interaction volumes were quite rare and very resource intensive until relatively recently. However, even sensor limitations do not explain why even with systems that use interpersonal spatial relationships only use them essentially for detecting when people form interactive groups, whether that is in order to interact with a third entity or between each other (e.g. to have a conversation). In Hall's book, the notion of interpersonal relationships is so much richer than that and it begs to be explored in more detail. Proxemics has been explored in interactive situations with humanoid robots [EH+13], virtual agents and augmented reality [OD+12], and when interacting with virtual characters [RAN05]. Additionally, Greenberg, Marquardt and their colleagues explore proxemics within physical interactive environments in their works [BMG10; WBG12; Wan12; MHG12; Mar13; Led14; LGB15; GL13] and their systems form a significant minority of the systems in this analysis. However, the focus of their exploration of proxemics seems to be more on extending Hall's concepts to interactions with objects and devices than exploring how Hall's concepts apply to humans within interactive environments. Therefore, there are still many opportunities for exploration in the application of proxemics.

Concept of Environment One detail that seemed common to a significant portion of the analysed systems was that it appeared that the authors of the systems did not consider the interactive environment in their designs. Arguably the focus of some of the systems was on specific interactions techniques or system designs. However, the fact that the interactive environment, or more generally Environment entities in any form, are generally not explicitly considered in system descriptions shows that interactions between Environments and other entities are likely under-explored.

Including the notion of the interactive environment in system design, considering how it impacts the interactions within it, as well as how the entity interactions can be enriched by interacting with the environment itself, presents researchers with many opportunities for further research. As an example of a possible opportunity, vehicular interaction could be a rich platform for systems that use relationships with Environment entities, depending on the classification granularity. If a vehicle is classified as an interactive environment, its spatial properties could play a significant role. For example, certain system functions could be disabled at motorway speeds for safety reasons or the amount and behaviour of surrounding cars could be used to extract contextual information and the vehicle could display additional environmental information to allow the driver to better deal with traffic.

Inter-Environment Relationships This discussion point is a corollary of the last mentioned one. As stated, relationships with Environment entities were not frequently used, or even openly considered by system designers. Environment-Environment relationships were never used. This is likely partly due to technical requirements as inter-Environment interaction usually requires a significant level of infrastructure support, which is often beyond the scope of research projects. Additionally, many research systems are highly focussed on a particular topic and therefore even though the system may have the potential to make use of Environmental relationships, the area is not explored. Using the vehicular interaction example from the previous point, using distance from fuel stations could be used to determine system-level power saving strategies to conserve energy if necessary.

Unintentional Interaction There is a clear gap in how the systems use intentionality of actions. There do not seem to be any system that makes use of *unintentional* actions. Moreover, a brief literature search reveals that very little research into unintentional interaction has been published in HCI venues to date, with the exception of Kuno et al. [KI+98].

Need for Higher Quality Descriptions Throughout the analysis, but especially when considering interaction measures, one issue hindered the analysis. The system descriptions (even when augmented by auxiliary material such as images and video footage) did not provide sufficient detail about the system to perform a detailed analysis. Replication is frequently highlighted as a concern with publications. Arguably, the level of detail required to replicate a system is not dissimilar to the amount of detail needed to completely analyse a system in detail. Therefore, if the information about systems was not sufficient to perform a relatively high-level analysis of the system, it is likely that fully replicating the systems would also present a challenge. On a positive note, availability of interaction models such as the IRE model as well as other descriptive frameworks provides researchers with tools to provide higher quality and more actionable information about systems.

3.12 Summary and Conclusions

This Chapter presented the results of an analysis of twenty-nine systems from literature together with four prototypes created for this thesis. The systems were analysed using the IRE model introduced in Chapter 2. After a brief introduction of each of the systems, their use of spatial relationships between interactive entities was examined. The main outcome of this part of the analysis was the general lack

of consideration of interactive environments as interactive entities. Where an Environment entity was part of the system, it tended to be utilised only in a limited fashion.

After the spatial relationships, the systems were analysed in terms of their more general interactional characteristics, such as modes of interaction or the cardinalities of interactive entities or the ranges, at which interaction took place. While some of the interactional characteristics showed significant similarities between systems (e.g. interactional modes were largely similar), other revealed notable variety as well as insight into possible design issues. This was most obvious with the analysis of action intentionality. Analysis of action intensity demonstrated an issue encountered throughout the analysis process as the amount of detail in the descriptions of published systems is generally inadequate to perform a thorough analysis.

However, even with the limited resources available, the analysis revealed a number of trends in existing systems as well as opportunities for future research. Moreover, it demonstrated the descriptive, comparative and generative abilities of the IRE model.

Chapter 4

Computer Vision Tracking Technologies for Spatially-Aware Interfaces

Researchers have previously explored a range of computer vision techniques for identifying and tracking facial features, including pupils, the eye area, nostrils, lips, lip corners and pose (e.g. [MCT09; YS+98; YK+02]). Once detected and tracked, such features can form parts of a multimodal interface (for an extensive survey see Jaimes and Sebe [JS07]).

Face-tracking has been used to realise "perceptual interfaces" that allow face movement to control games [WX+06] and 3D graphics interfaces [Bra98]. Head movements can be translated into control variables, which can then be used to control a mouse pointer. Researchers have also investigated how to detect and interpret user's gaze. Vertegaal et al. [VD+02] use a custom *EyeContact* sensor coupled with a mobile phone to detect whether the user is engaging in a conversation. Later, Dickie et al. [DV+05] propose a range of scenarios on how to use the same EyeContact sensor to adapt mobile applications depending on whether the user is looking at the mobile phone display or not. A related technique is to use gaze-gestures for mobile phone interaction [DDS07].

When it comes to detecting distance for interactive purposes, several computer vision based approaches have been used. A large number of recent systems make use of the Kinect sensor (or its variants), whether mounted in a standard horizontal manner (e.g. [BD11; SMB13; BB+14]), or vertically — usually on the ceiling (e.g. [MHG12; WB10; HR+14]). The next most commonly used systems are Vicon (e.g. [VB04; BMG10; WBG12; BL+14]) and OptiTrack (e.g. [JS+13; RJ+13]). Beyond that, some researchers develop their own computer vision based techniques using single RGB [HD08] or stereo cameras [RLW97; KH+00].

The systems created for this thesis use computer vision based tracking systems as they provide good potential for use on commonly available hardware. This chapter starts by providing an overview of existing computer vision based distance tracking systems. This is followed by a presentation of an initial exploration into the use of standard feature classifiers for estimating distance of a person from a camera. The strengths and weaknesses of this approach are evaluated and the results are used to develop a distance estimator using a single RGB camera. The distance estimator is evaluated for accuracy of distance estimation as well as the reliability of its detections.

This work is then extended in two ways. Firstly, the distance estimator is modified to provide a binary measure of orientation to support *DiffDisplays* — a system described in detail in Chapter 7. The next extension enables support for multi-user tracking of up to four simultaneous users as well as higher accuracy distance estimations using a combination of an RGB camera and a depth camera. This version of the tracking system is used in Chapter 6 as part of a prototyping toolkit. The chapter concludes with a set of recommendations for designers on the use of computer vision based techniques for distance estimation in interactive systems.

System	Sensor Position	Maximum Detected Distance
Lean & Zoom [HD08]	Environment (camera) + Person (markers)	unknown (likely <2 m)
EasyLiving [KH+00]	Environment (cameras)	unknown (likely <5 m)
Smart Kiosk [RLW97]	Environment (cameras)	unknown (likely >4.5 m)
KinectV1	Environment (cameras)	up to 4 m
KinectV2	Environment (cameras)	up to 4.5 m
Vicon	Environment (cameras) + Person (markers)	up to 12 m
OptiTrack	Environment (cameras) + Person (markers)	up to 15 m
CV Only	Environment (camera)	up to >5 m
KinectV1+CV	Environment (cameras)	up to 5 m

Table 4.1: Comparison of Existing Distance Sensing Systems, the *CV Only* and *KinectV1+CV* systems are introduced later in this chapter.

4.1 An Overview of Existing Room-level Computer Vision Distance Tracking Systems

Both off-the-shelf and custom designed tracking systems have been used in interactive systems for distance based interaction. The following systems are a small sample of the best known systems not using computer vision techniques — *Active Badge* [WH+92; HH94], Bats [HH+02], RADAR [BP00; BPB00], Cricket [PCB00], or commercially available systems such as the ultrasonic InterSense products. However, this overview concentrates of tracking systems primarily using computer vision techniques.

Table 4.1 shows a visual overview of the systems. The tracking systems tend to have two main distinguishing features — whether they require augmentation of the tracked persons with markers and their maximum detected distances. Using these two features, it is possible to observe two main groups of tracking systems — long range, high accuracy systems requiring human augmentation, and shorter range, generally lower accuracy systems that to not require human augmentation. There is one tracking system, which does not fit into either group. The tracking system used in *Lean & Zoom* [HD08] both uses markers and as described has a very short maximum range. However, even this tracking system functions sufficiently well to validate the research contributions in the publication.

The group of long-range, high-accuracy tracking systems contains two marker based systems using a number of cameras in the environment and passive markers placed on the object to be tracked. Both Vicon and OptiTrack allow for comparatively long range tracking, but the systems come at a cost. Both of the tracking systems are relatively expensive and require a number of cameras to be deployed throughout the tracked volume. Additionally, both of the systems require careful calibration before they can be used. Moreover, since both systems require any tracked entity to be augmented with markers, this increases the barrier to interaction and likely to influences how natural any interactions feel to the system users. However, the use of markers makes it possible to track any entity, including objects or animals. Additionally, any body part can be instrumented, opening a number of interactive possibilities.

The second group of tracking systems tend to offer tracking only within a shorter range (around 4–5 metres). However, the relative range disadvantage is offset by not requiring any augmentation of the people interacting with the system. This creates opportunities for walk-up interactions or interactions, where markers would interfere with the user experience. The lack of markers does come with additional constraints. The tracking systems are generally tailored for tracking humans and do not offer any object tracking capabilities. The markerless tracking systems diverge in their approach to tracking. Each uses computer vision techniques but the specifics differ. The Kinect

(both versions) uses a depth sensing camera. The first Kinect uses the deformation in a projected speckle pattern to create a depth map, whereas the second version of the Kinect utilises a time of flight camera. The *EasyLiving* and *Smart Kiosk* tracking systems use stereo cameras. Lastly, there are the two tracking systems introduced in this chapter. The CV Only version of the system uses a single RGB camera and exploits a known distance between the tracked person's eyes, while the Kinect-CV version of the system combines a single RGB camera with a depth camera of the first version of the Kinect. These two tracking systems will be the focus of the rest of this chapter.

4.2 Initial Approach to Markerless Tracking

In order to utilise a person's distance from an object it is necessary to be able to measure it accurately, preferably doing so using inexpensive commodity hardware. Previously, the most closely related systems to measuring viewing distance, or detecting whether the user is looking at the screen, have relied on custom hardware [DV+05] or markers [HD08]. In contrast, the primary approach taken here explores a system designed to work with commodity hardware and no markers.

4.2.1 General Approach

The detection system uses the OpenCV computer vision library[1]. This decision was based on an evaluation by Castrillón-Santana et al. [CSDS+08] who use OpenCV to perform feature detection for facial detection. In this case, OpenCV is used to detect faces and eyes for distance estimation. The feature detection algorithm in OpenCV is an implementation of the Viola-Jones feature detection algorithm [VJ04].

The two different classifiers used for this initial approach were chosen based on observations of the available raw data, the data needed for distance estimation, and on the required processing time. Each of the classifiers have their own strengths and weaknesses in different experimental settings.

The One-Stage classifier uses a Haar cascade from the OpenCV library trained to detect eye-pairs (this classifier is referred to as EP1 in [CSDS+08]). The classifier provides two advantages. First, it reduces the risk of recognition errors as the eyes in conjunction with the nose form a more complex visual pattern than only a single eye alone, and therefore more visually distinct object than the eyes do on their own. Second, the Haar cascade is trained primarily on frontal face images. Because of this the eye-pair is not detected when the person is not directly facing the camera (and by extension, when the person is not looking at the screen).

The Two-Stage classifier uses the same underlying technique as the One-Stage classifier. However, instead of using a single cascade it uses two. The first cascade detects faces (this cascade is referred to as FAT in [CSDS+08]). The area of the image where the face is located is then used as an input area for the second cascade. The second cascade is the same as in the One-Stage classifier. The decision to test the Two-Stage classifier was caused by the frequency of misclassifications by the One-Stage classifier. By ensuring that the classifier searches for the eye-pair in an area where eyes are expected, the risk of recognition errors can be reduced. The disadvantage of the two-stage approach is that it is more resource intensive.

The parameters of the OpenCV function for detecting objects using a Haar cascade were set as follows. First, the scale factor is 1.2, which means the search window size is increased by 20% between scans of the image. Second, the number of neighbouring rectangles that make up an object were set to one. This means that groups of less than one rectangle are rejected. Third, the system uses a Canny edge detector to reject image regions that contain too many or too few edges and thus cannot contain the object being detected. The net result of this is a speed increase.

[1] http://opencv.org

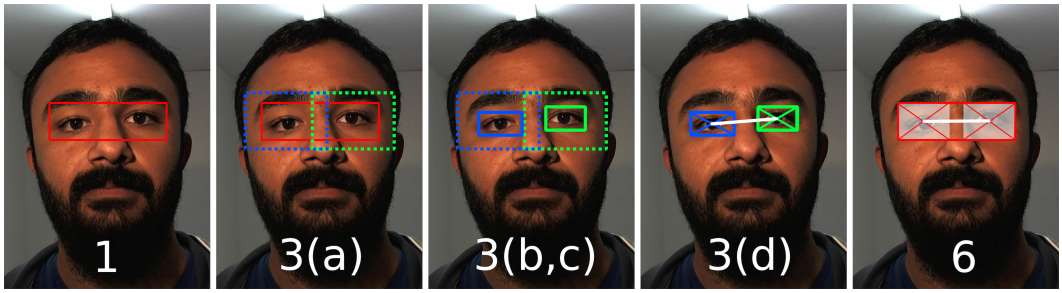


Figure 4.1: A visual representation of the different stages of the distance detection algorithm. The numbering of the images corresponds to the stages of detection.

4.2.2 Algorithm for Estimating a Person's Distance from a Camera

The distance estimation algorithm consists of two parts: image analysis and distance computation. The algorithm for image analysis is based on the One-Stage algorithm with some additional steps to increase accuracy. The main steps of the algorithm are visually illustrated in figure 4.1.

For a given image, the algorithm works as follows.

1. Find the areas of interest (the rectangles containing individual eye-pairs) using the One-Stage/EP1 classifier.
2. If no eye-pairs are found, fail and take no action.
3. Otherwise, for each detected eye-pair:
   a) Partition the area of interest horizontally into two parts, adding 20% horizontal and 60% vertical padding to each of the two sub-areas of interest.
   b) Scan the left sub-area of interest for potential rectangles containing the left eye using the LE classifier and select the largest one.
   c) Scan the right sub-area of interest for potential rectangles containing the right eye using the RE classifier and select the largest one.
   d) If both sub-areas of interest contain a successfully detected eye, find the position of the pupils. Each pupil is located in the centre of the containing rectangle for its respective eye. Then, compute the person's distance from the camera according to Formula 4.1.
   e) Store the location of the pupils and the sizes of the containing rectangles for the eyes together with the computed distance.
4. From all the possible eye-pairs, select the one that has both eyes detected and for which the sizes of the rectangles containing each eye closely match each other.
5. If such an eye-pair is found, report a full detection, including the distance between the pupils within the area of interest in pixels.
6. If no such eye-pair is found, fail gracefully by noting a partial detection with the location of the largest eye-pair found as the most likely eye-pair. This is computed using the same method but instead of using the position of the pupils, the rectangle occupied by the eye-pair is partitioned into three horizontal segments (40%, 20%, 40% - see the right-most image in figure 4.1 for an example) and the centres of the two 40% segments are used as the distance points.

The distance is computed based on the following formula:

$$D = \frac{\frac{D_e}{2}}{\tan(\frac{D_p}{2} \times \frac{C_a}{C_r})} \tag{4.1}$$

where
D – the distance of participant's eyes from the camera in millimetres
D_e – participant's pupil distance in millimetres (measured during calibration)
D_p – pupil distance in pixels (as computed in step 2(d) of the algorithm)
C_a – camera's horizontal angle of view in degrees
C_r – camera's horizontal resolution in pixels

Intuitively, the formula works as follows. The pixel distance is taken as a base of an isosceles triangle. The length of the altitude/median between the base (participant's eyes) and the vertex opposite the base (the camera) is computed. This is done using a trigonometric function on a right-angle triangle formed by the altitude/median of the isosceles triangle, a half of the base of the isosceles triangle and one of the sides of the isosceles triangle.

In this form the algorithm is strict in its processing and only proceeds to distance computation when the most precise position of the individual eyes is available. However, under some circumstances it may be beneficial to provide a rough distance estimate even when more precise data is unavailable. In that case, the algorithm could fail gracefully by computing the distance based on the eye-pair rectangle rather than specific eyes. While this will be inherently less precise, the precision can be sufficient for a rough estimate. This graceful fallback is demonstrated in the last image in figure 4.1.

4.2.3 Evaluation 1: Accuracy of Detections and the Effect of Changing Head Orientation

In order to determine the suitability of the distance estimation algorithm as the basis of a technological platform for conducting experiments on prototype distance-aware interfaces, it was necessary to evaluate its accuracy and robustness. This first evaluation primarily tested the detection accuracy of the underlying classifiers, as well as their specific combination in the algorithm. An additional goal of the evaluation was to validate the choice of the One-Stage eye-pair classifier as the classifier of choice for the first stage of the distance estimation algorithm. The evaluation was conducted in a controlled laboratory setting. Two input devices were examined: a desktop computer equipped with a screen with a built-in web camera (Desktop) and a mobile phone with a front-facing camera (Mobile).

Method

The experiment investigated two settings (Desktop and Mobile), two classifiers (One-Stage and Two-Stage), and two viewing distances.

For the experiment, four participants from a local university campus were recruited. Their ages ranged between 23 and 29. All were male. Two participants wore glasses while the other two used neither glasses nor contact lenses.

The Desktop setting used an 20″ 2006 iMac with an integrated web camera. The Mobile setting used an iPhone 3GS and its integrated camera. Since one of the aims of the experiment was consistency between the settings and the iPhone's camera was limited to VGA resolution (640×480 pixels), the same image resolution was used for both of the settings.

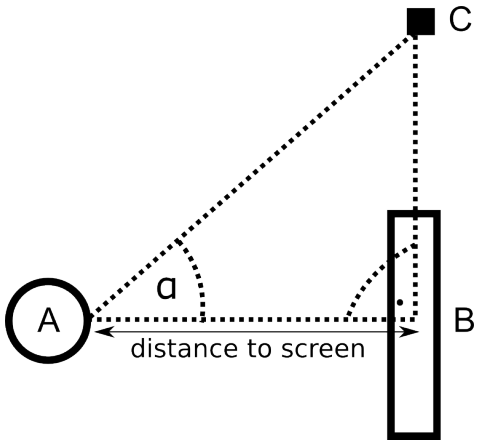


Figure 4.2: A diagram illustrating the positioning of the user (A) , the display (B) and one of the targets (C) for Evaluation 1.

For the Desktop setting, the two distances tested were 90 cm and 60 cm. These distances roughly corresponded to distances at which the participants were comfortable viewing the screen. These distances were roughly equivalent to horizontal viewing angles of 36° and 53°, respectively.

For the Mobile setting, the viewing distances were 60 cm and 30 cm. 60 cm roughly corresponded to holding the mobile phone at an arm's length, while 30 cm roughly corresponded to holding the mobile phone at a closer and more comfortable distance.

Markers were positioned around the room to provide participants with visual targets to make the gaze directions as consistent between participants as possible. They were all also advised to move their head, not just their eyes, as the important factor was not the movement of the eyeballs but rather the movement of the head, as this changes the visual appearance of the eye-pair the most from the point-of-view of the camera.

Four markers were positioned so that they surrounded the participant. Figure 4.2 shows a diagram for how a prototypical marker was positioned. Point *A* was the middle of the participant's nose. In the Desktop setting point, *B* was the centre of the screen. In the Mobile setting, point *B* was the centre of the mobile phone screen. Point *C* was the position of a marker. There were five markers in total, one to the left of the participant (*left*), one to the right (*right*), and one positioned in the centre of the display (*centre*). The last two markers were positioned above and below the *centre* marker (*up* and *down*, respectively).

In the Desktop setting the distance to the screen was 75 cm. The angle α was 30° vertically for *up*, 60° vertically for *down*, and 70° horizontally for *left* and *right*. In the Mobile setting the distance to the screen was 45 cm. The angle α was 60° vertically for *up* and *down*, and 70° horizontally for *left* and *right*. The angles used for marker positioning were measured from the participant's seating position and corresponded to the angular difference from the centre of the display (points *A* and *B* in Figure 4.2, respectively).

In each of the four different conditions, every participant was instructed to look at the direction markers in the following sequence: *centre, left, centre, up, centre, right, centre, down, centre*. The participant was times so that they looked at each marker for 10 seconds. This means that participants looked directly at the screen 55% of the time. The other 45% of data enabled finding out how well the classifiers handled different gaze directions.

Setting	Classifier	Distance	Accuracy	TP	TN	FP	FN
Mobile	One-Stage	Near	67.08%	43.25%	23.84%	18.17%	14.74%
		Far	72.79%	48.95%	23.84%	14.77%	12.44%
	Two-Stage	Near	68.69%	27.76%	40.94%	0.88%	30.43%
		Far	62.91%	27.41%	35.49%	6.02%	31.07%
Desktop	One-Stage	Near	76.87%	56.49%	20.38%	22.94%	0.19%
		Far	72.96%	55.08%	17.88%	26.30%	0.74%
	Two-Stage	Near	82.54%	54.50%	28.04%	15.30%	2.17%
		Far	69.74%	47.47%	22.26%	21.48%	8.79%

Table 4.2: Comparison of accuracy of classifiers. In order to provide more granularity to the data, the percentages of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) are also included.

Results

In total, 24 minutes of video data were collected. Each participant generated 90 seconds of video data for each configuration (six minutes in total). The video stream was then processed by the One-Stage and Two-Stage classifiers to generate information about the eye-pairs they detected. The video stream was recorded at 30 frames per second and the classifiers processed every single video frame.

The region that enclosed the eye-pairs was saved whenever the classifiers detected them. The positions and sizes of the bounding boxes of every single eye-pair were also saved to form the basis of a distance profile. All of the saved data was timestamped so that each of the samples could be easily identified and synchronised with any other data from the same participant.

A human judge marked the image data outputted by the classifiers into one of three categories:

- Valid: the classifier believed that the participant was looking at the screen and the human judge agreed.
- Edge case: the classifier believed that the participant was looking at the screen and the human judge disagreed. However, the eye-pair was captured properly. It was only the direction of the gaze that was wrong.
- Invalid: a complete misclassification. The classifier believed the participant was looking at the screen, when in fact there were no human eyes present in the sample image.

Each video stream was then coded by a human judge with information about the start and end times of the experiment and information about whether or not the participant was looking at the screen. The same judge also marked the parts of the stream, where it was impossible to tell where the participant was looking at. Additional information about gaze direction was encoded as well to allow an analysis of how well the classifiers handle the different gaze directions.

Table 4.2 shows how accurate the classifiers were in detecting whether the participants were looking at the screen or not. Overall, the One-Stage classifier performed better in the Mobile setting than the Two-Stage classifier. The Two-Stage classifier performed better in the Desktop setting, but not by a large margin.

Table 4.3 shows how the classifiers handled different gaze directions. In the Mobile setting the One-Stage classifier had an accuracy higher than 90% along the lateral axis (centre, left, right). However, the accuracy for up and down movements was low (see the second and third column in table 4.3. The Two-Stage classifier performed notably better in every single gaze direction except for the centre. This was because the Two-Stage classifier had difficulties in detecting faces at close distances.

Classifier	Centre	Up	Down	Left	Right
Mobile Environment					
One-Stage	91.06%	37.40%	0.00%	95.00%	92.42%
Two-Stage	49.95%	96.70%	69.08%	98.46%	99.04%
Desktop Environment					
One-Stage	100.00%	0.00%	4.67%	94.93%	76.42%
Two-Stage	90.81%	0.00%	48.29%	95.93%	87.12%

Table 4.3: Comparison of accuracy of classifiers for gaze direction

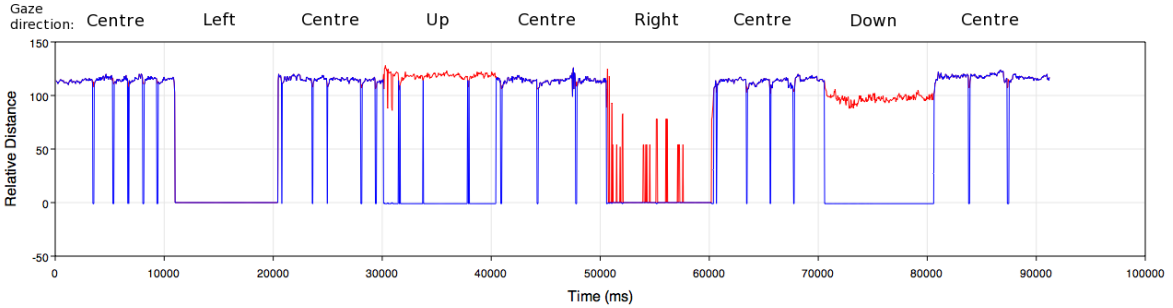


Figure 4.3: A comparison of valid and invalid classifications in the Desktop setting for a typical participant. A perfect classifier would follow the blue line. The occasional dips during periods when detections were expected correspond to the participant blinking and thus were not considered to be incorrectly classified.

In the Desktop setting the overall performance of the classifiers was similar for all gaze directions, except when participants were looking down. The Desktop setting faired much worse in detecting up and down movements than the Mobile setting. This is probably because the angular difference between the centre point of the screen and the specific markers participants were gazing at when looking up or down was less for Desktop than Mobile.

Figure 4.3 demonstrates the difference between correctly and incorrectly classified data. It was produced by the One-Stage classifier. The blue line shows how a perfect classifier would classify the data. The red line shows the actual classification outcome. If the classifier is always accurate the red line will always be hidden behind the blue line. The red line is only visible in the figure when the classifier has made a classification error. The experimental sequence can be easily seen in the figure. Each of the roughly 10,000 ms segments shows different directions of gaze (as labeled above the data). The figure shows that for the centre and left gaze directions, the classifier performs very well. However, for small parts of the right direction segment and all of the up and down segments, the classifier completely misclassified the data.

In conclusion, this evaluation demonstrated that the One-Stage classifier produces significantly higher amounts of false positives, while the Two-Stage classifier is more biased towards producing false negatives. Since the primary algorithm contains additional measures to verify the validity of detections and the performance penalty of the Two-Stage classifier is sometimes approximately an order of magnitude, the choice of the One-Stage classifier as the primary classifier is justified.

4.2.4 Evaluation 2: In-The-Wild Eye-Pair Detection

Evaluation 1 examined the ability of the classifiers to detect the presence of gaze in a controlled environment by detecting the participant's eye-pair. The purpose of the second evaluation was to

Figure 4.4: Simulating a front-facing camera in early 2011.

test how well the classifiers would work under more real-life conditions through a task involving a mobile phone in a combination of outdoor and indoor environments.

Method

The scenario for this evaluation was to simulate a person walking somewhere while simultaneously trying to perform a task on a mobile phone. *Sudoku* was chosen as the task because most people are familiar with the game, the rules are simple, and it does not require any complex mathematical knowledge in order for a person to be able to play it. Moreover, it is independent of language and culture, which means that any linguistic or cultural influence on the participant's ability to complete the task can be discounted. The task is also time consuming enough to last the whole length of the experiment, and it can be easily repeated in case more time is needed for data collection. Seven participants were recruited for this experiment, their ages ranged between 19 and 29. Four of the participants were female, three male. Four participants wore glasses and the other three participants wore neither glasses nor contact lenses.

The same iPhone 3GS mobile phone that was used in Evaluation 1 in the Mobile condition was used for data collection in this evaluation as well. However, in order to simulate a mobile phone with a front-facing camera, it was necessary to glue two separate mobile devices together. The iPhone 3GS was turned 180° and glued to an iPod Touch with an offset so that the camera on the iPhone was facing the participant. The screen of the iPhone was covered in order to make sure that the touch capability was disabled to avoid accidental touch input. Figure 4.4 shows the resulting mobile device. All of this meant that the resulting device was somewhat less firm and easy to hold than a single device would have been. However, none of the participants reported any trouble in using it.

First, participants had familiarised themselves with the equipment and demonstrated that they knew how to play Sudoku. Thereafter they were taken on a roughly eight minute journey around an

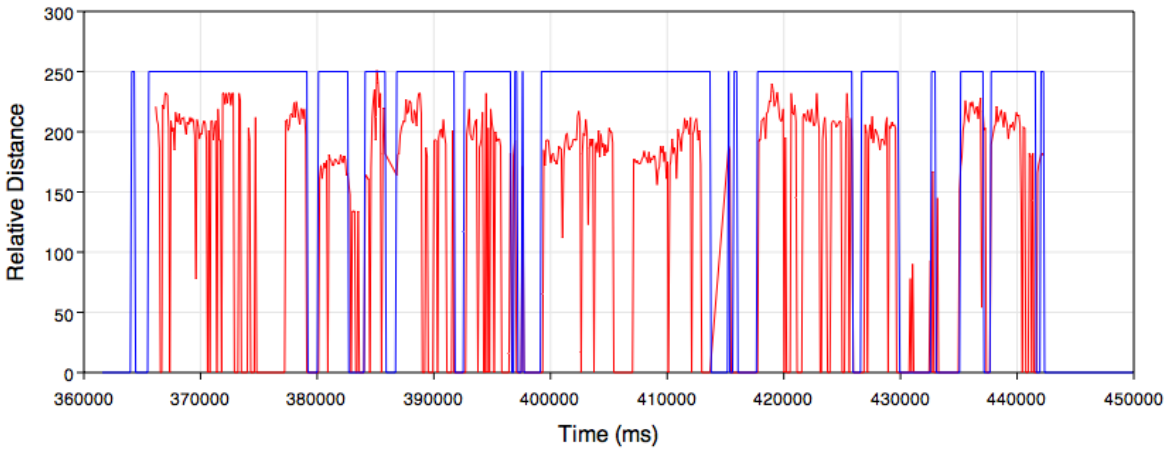


Figure 4.5: A segment of a distance profile for a typical participant in Evaluation 2.

indoor laboratory environment and a surrounding outdoor area. On the journey, they followed a guide. The guide ensured that every participant would follow the same path and at a similar pace.

The path presented the participants with a range of environments (indoor, outdoor) and obstacles (navigating around furniture, opening/closing doors, ascending/descending stairs). It also exposed the equipment to a range of lighting conditions (uneven artificial lights, clear/cloudy sky outdoors).

Results

Each of the participants generated approximately eight minutes of data. In total, nearly an hour of video was collected. The video stream was processed by the One-Stage and Two-Stage classifiers to generate the same type of output data as in Evaluation 1. However, the only every third frame was processed (giving an effective frame rate of 10 frames per second) in order to reduce the number of samples for processing. The image data was marked by a human judge into the same classes as in Evaluation 1.

Video stream segments were annotated into three classes: a) participant looking at the display, b) participant not looking at the display, or c) insufficient visible data to determine whether the participant was looking at the display or not. An example of an instance of the latter class was if the face so dark that the eyes could be distinguished.

The One-Stage classifier resulted in a much higher accuracy (46.1%) than the Two-Stage classifier (16.8%). This was because the Two-Stage classifier often failed to detect the face. This resulted in an immediate classification failure. However, note that the failure to detect the face was often caused by the fact that the full face of the participant was often not present in the frames due to the narrow field of view of the camera and the participant's closeness to the camera.

Figure 4.5 shows a representative data segment for the best-performing One-Stage classifier for a typical participant. The red line shows relative distance as it was detected by the One-Stage classifier. The blue line is a binary function that shows ground truth about whether the participant was looking at the screen or not (shown as values of either 250 or zero respectively). As is evident in the figure, even though this classifier performed reasonably well the results in Evaluation 2 were worse in comparison to the results in controlled conditions in Evaluation 1 (see Figure 4.3 for visual illustration). The One-Stage classifier once again appeared to be the better choice, although here the primary reason was that it required a smaller part of the face to be visible in order to deliver useful results. The One-Stage classifier also seemed to be somewhat more resilient in the face of varying lighting conditions.

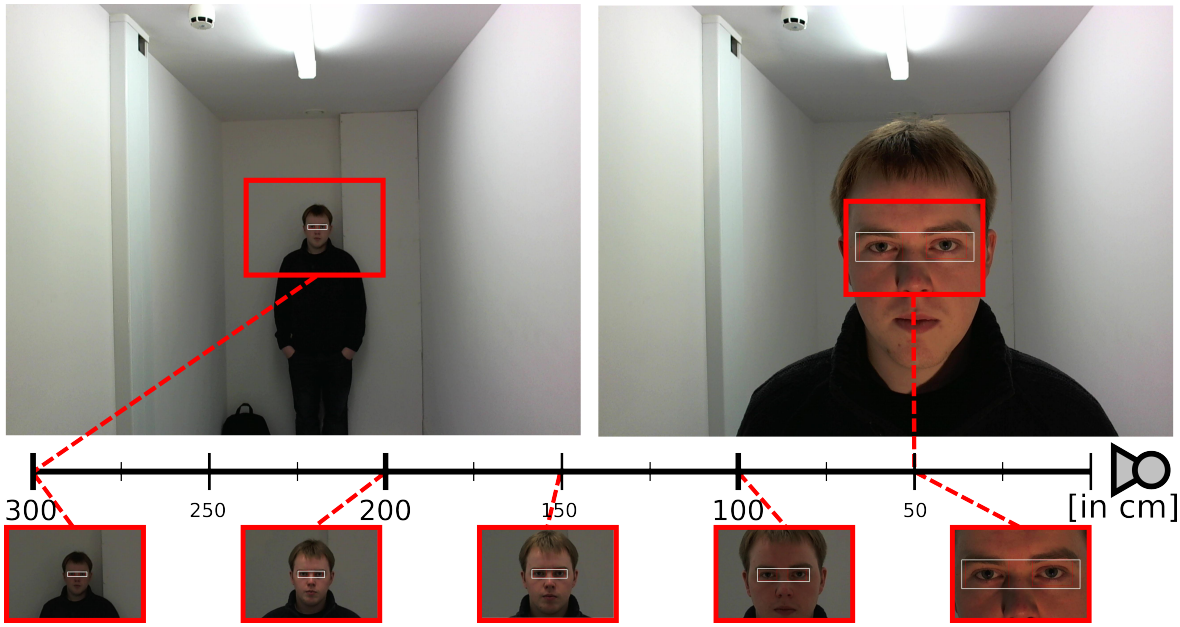


Figure 4.6: An illustration of the difference in eye sizes with distance changes. All images in red rectangles are 100% crops.

Considering the accuracy disparity between the One-Stage and Two-Stage classifiers under the varying lighting and usage conditions in this evaluation, it was clear that the bias of the Two-Stage classifier made it largely an impractical choice for the distance estimation algorithm. This further validated the choice of the One-Stage classifier. However, this study also demonstrated the difficulty in employing computer vision based distance estimators in un-controlled settings without additional modifications. Part of the discussion in Section 4.2.6 provides suggestions for mitigating some of the factors that pose a challenge for computer vision algorithms in these settings.

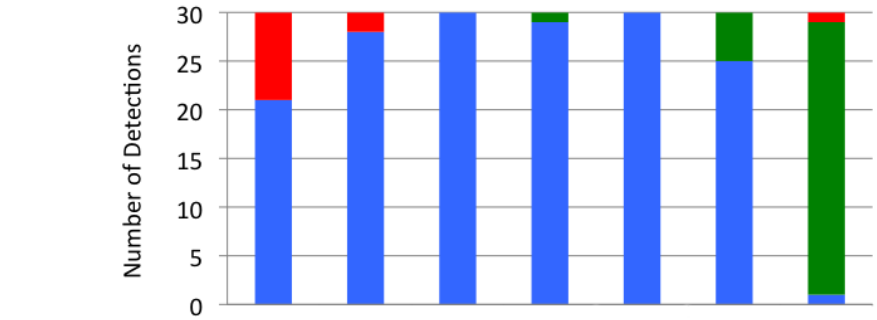
4.2.5 Evaluation 3: Estimating Absolute Viewing Distance

With the accuracy of the underlying classifiers established and the classifier choice for the distance estimation algorithm validated, the next evaluation target was to establish the algorithm's ability to estimate a person's distance from the camera. Evaluation 3 investigates how well a person's distance from the camera can be estimated using the initial approach under controlled, but somewhat variable lighting conditions.

Method

For this evaluation, I recruited 15 participants. Their ages ranged between 19 and 26 (mean age 21.93). Three participants were female and twelve were male. None of the participants wore glasses.

I chose seven distances for evaluation: 25 cm, 50 cm, 75 cm, 100 cm, 150 cm, 200 cm and 300 cm. Distances up to 100 cm are more fine grained as they are more representative of the distances common for interacting with mobile phones and desktop computers. Distances between 100 cm and 300 cm are less fine grained for two reasons. First, these are less common interaction distances in desk-based systems. Second, the ability of a computer vision algorithm to detect an object in an image depends on the size of the object in pixels. With increasing distance, the size of the eyes will



Distance (mm)	250	500	750	1000	1500	2000	3000
No Detection	9	2	0	0	0	0	1
Partial Detection	0	0	0	1	0	5	28
Full Detection	21	28	30	29	30	25	1

Figure 4.7: Distance detections at medium resolution (1600×896 pixels).

decrease geometrically. This means that the difference in size is less pronounced at longer distances and therefore small distance intervals would not contribute much to the results.

A consumer-grade webcam (Logitech C910) was mounted on a tripod and the chosen distances were measured from the position of the camera in a straight line and marked up on the floor. Since computer vision algorithms tend to be influenced by lighting conditions, a desk lamp was positioned so that it would provide additional light for short distances as the ambient lighting was rather dim (only one fluorescent light on the ceiling).

Due to the fact that the angle of view of the camera changed depending on the chosen resolution, two different resolutions were tested: a 4:3 ratio high resolution (2592×1944) and a 25:14 ratio medium resolution (1600×896). The horizontal angle of view of the camera was 62° for the standard 4:3 ratio resolution and 70° for the wide 25:14 ratio resolution.

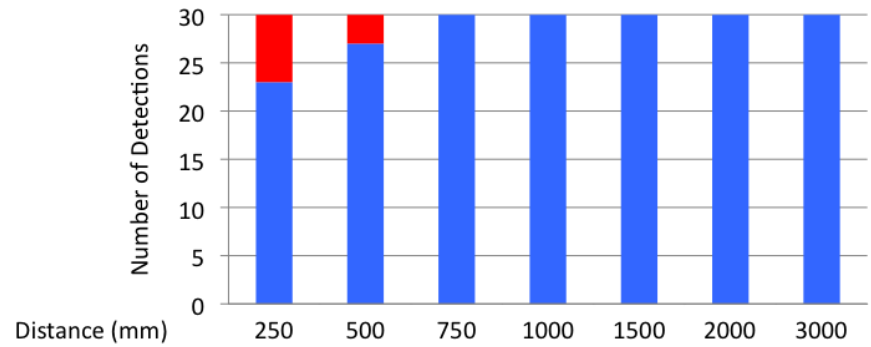
During the evaluation, each participant was positioned on each of the distance marks on the floor in turn. A ruler positioned perpendicular to the floor was used to ascertain that the eyes of the participant were positioned above the distance mark. Two still pictures from the live camera video were captured. After the capture, the participant would be lead to the next distance mark on the floor. When all the still pictures for the seven distances had been captured, the procedure was repeated for the second resolution.

Results

A total of 420 still images were collected for this evaluation (two images for each of the seven distances at two different resolutions for each of the fifteen participants).

For some of the images, the auto-focussing mechanism of the webcam failed to focus properly (four images). There was also some softness in most images captured at 1600×896. The softness was probably due to in-camera interpolation for wide-ratio resolutions as softness was observed at other wide-ratio resolutions as well. Due to the lighting setup in the room, many of the images with participants standing far away (at the 300 cm distance) did not have much light illuminating the participants. For images of participants standing close to the camera, the secondary light source sometimes resulted in areas of shadows on the faces of some participants.

Additionally, I also found four images, in which a participant was blinking, as well as further 24 images where a participant was not looking directly at the camera. However, all the captures were



Distance (mm)	250	500	750	1000	1500	2000	3000
No Detection	7	3	0	0	0	0	0
Partial Detection	0	0	0	0	0	0	0
Full Detection	23	27	30	30	30	30	30

Figure 4.8: Distance detections at high resolution (2592×1944 pixels).

included in the processing rather than excluding them as they better reflect real-life situations and conditions of a system deployed in general use.

Figures 4.7 and 4.8 show that the algorithm very rarely failed to completely identify the eyes (5.71% for 1600×896 and 4.76% for 2592×1944 over all the distances). Failed detections arose at very short distances (25 cm or 50 cm), or at the maximum distance (300 cm).

Partial detections only occurred at the lower resolution (1600×896) and at long distances (200 cm and 300 cm). This indicates that at long distances the lower resolution provides sufficient information to detect the eye-pair, but insufficient information to determine the accurate position of each individual eye.

Figures 4.9 and 4.10 show the accuracy of distance estimations for all the complete detections at the different distances. The image set for each distance was 30 images. The distance estimations are from full detections in each of the image sets. Except for the distance at 300 cm at the 1600×896 resolution, the estimation for each distance is based on at least 21 detections, and in most cases almost 30 detections.

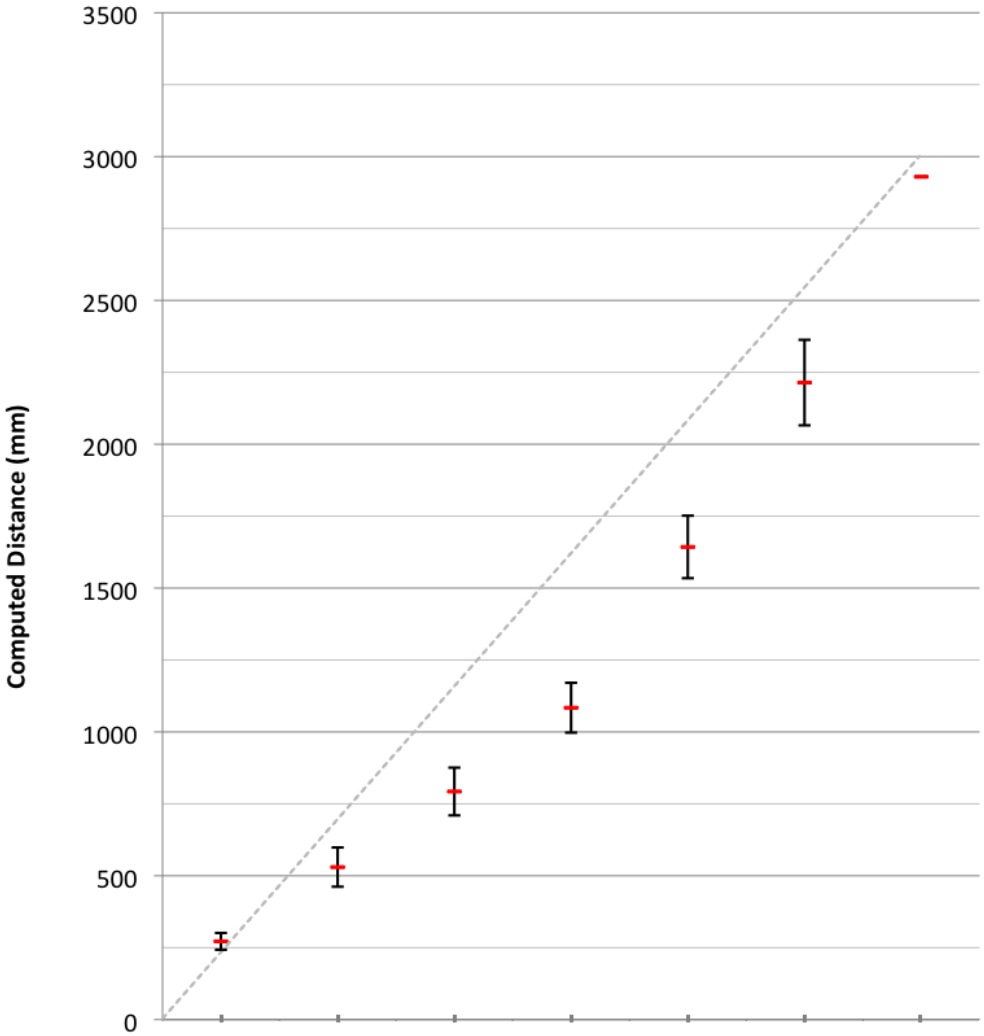
The mean estimated distances have a worst case relative error of no more than 11%.[2] The worst estimation error for the lower resolution was an error of 46 cm at a distance of 200 cm. The worst estimation error for the higher resolution was 36 cm at a distance of 300 cm. In general, estimation errors increase with distance as the pixel resolution becomes more important as the distance increases.[3] It is important to note that the tips of participants' feet were aligned with the distance markers, rather than their eyes. Since most of the participants' eyes were between 0 and 10 cm behind the tips of their feet (at the extremes), a distance estimation error of up to 10 cm (with a mean of approximately 5 cm) is expected.

4.2.6 Discussion

The three initial evaluations revealed that it is indeed possible to detect eye-pairs that can then be used to estimate the person's distance from the camera and that such an estimator can be built using built-in and consumer grade cameras on desktops and mobile phones. Evaluation 1 validated the

[2]The relative error is the ratio between the estimation error and the actual distance.

[3]At 300 cm, a pupil distance of 63 mm corresponds to approximately 50 pixels for a horizontal resolution of 2592 pixels and only 27 pixels for a horizontal resolution of 1600 pixels.

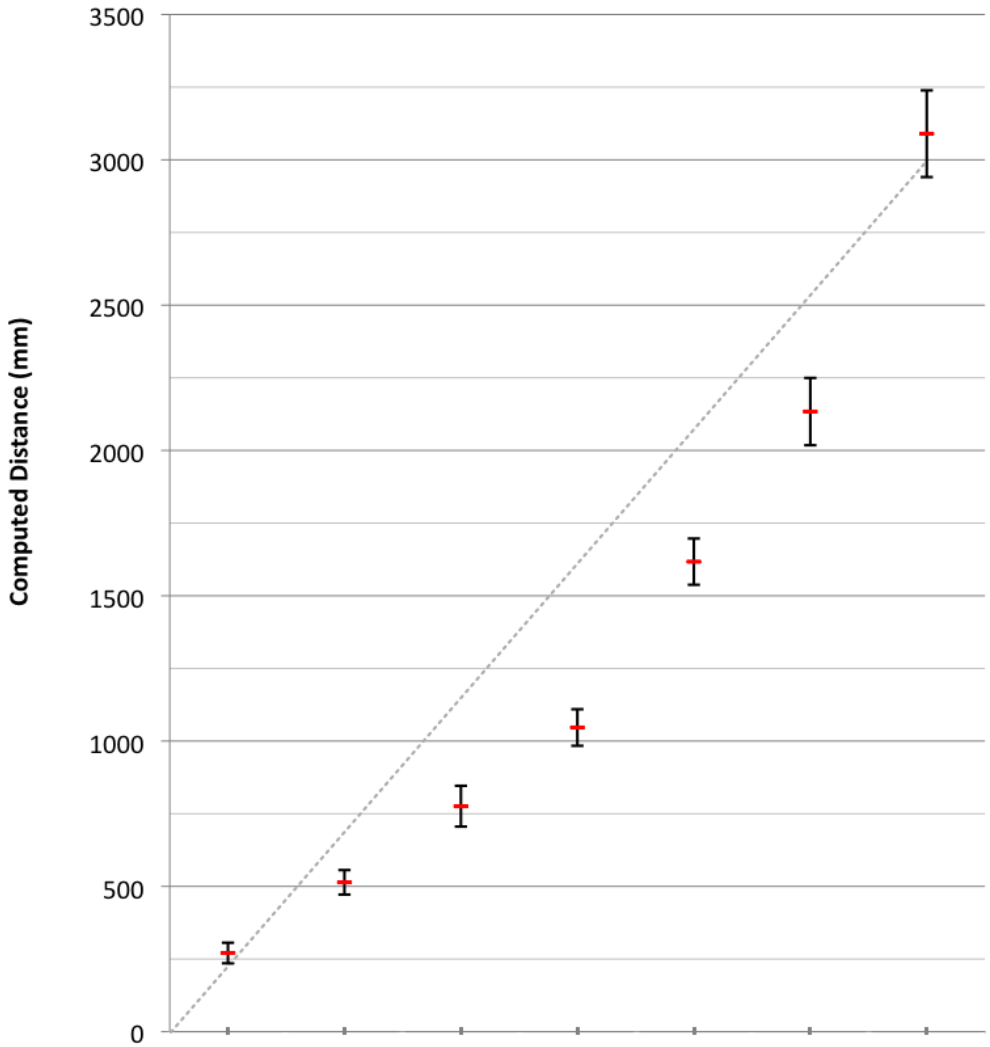


Expected Distance (mm)	250	500	750	1000	1500	2000	3000
Maximum Computed Distance	327	801	1052	1348	1921	2460	2930
Minimum Computed Distance	219	464	689	950	1410	1902	2930
Average Computed Distance	272	530	793	1084	1643	2214	2930

Figure 4.9: Results of absolute distance evaluation at medium resolution (1600×896 pixels). Error bars show standard deviation.

choice of the One-Stage classifier for the distance estimation algorithm due to its lower bias towards producing false negatives and for performance reasons. Evaluation 3 further demonstrated that in a controlled environment with relatively constant, if not uniform, lighting conditions it is possible to estimate a person's distance from the camera with a sufficiently high accuracy for the distance estimation method to be used as an experimental platform.

However, Evaluation 2 demonstrated that accuracy of the classifiers decreases dramatically in heterogeneous use such as scenarios with varying lighting conditions. When participants were walking around in the second evaluation, some participants positioned the camera so that their faces were not fully contained in the camera image. This resulted in classification failures, in particular



Expected Distance (mm)	250	500	750	1000	1500	2000	3000
Maximum Computed Distance	350	641	986	1167	1811	2319	3364
Minimum Computed Distance	217	470	704	935	1476	1885	2808
Average Computed Distance	271	514	776	1047	1617	2134	3090

Figure 4.10: Results of absolute distance evaluation at high resolution images (2592×1944 pixels). Error bars show standard deviation.

for the Two-Stage classifier. Cameras with wide-angle lenses could potentially reduce this problem. Also, some participants used glasses while others did not. At certain angles glasses caused reflections that were seen by the camera in such a way that the reflection concealed their eyes and thus caused misclassifications. The rims of the glasses sometimes also obscured the eyes. Last, in some cases participants' long hair sometimes obscured their faces or eyes.

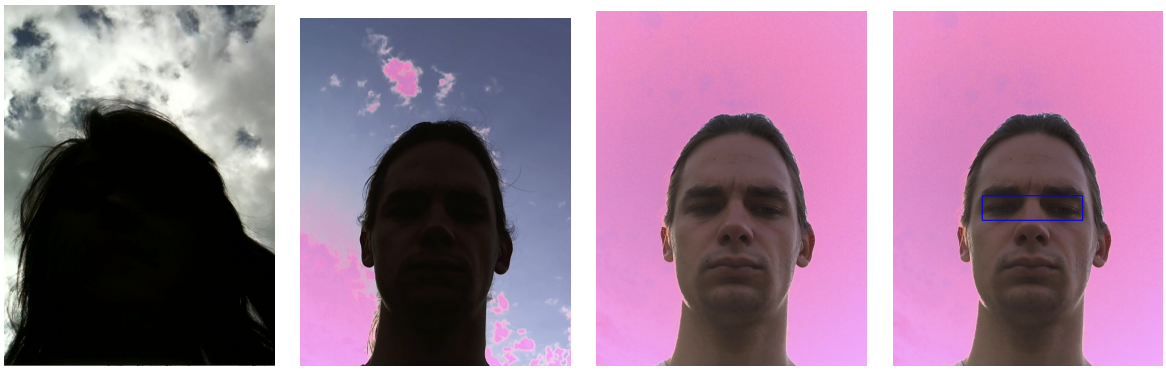


Figure 4.11: Four images demonstrating exposure problems. Taken from left: 1) a frame from Evaluation 2 described in section 4.2.4, 2) a sample frame from a webcam taken with automatic exposure settings, 3) a sample frame of the same subject under the same conditions as the second frame, but this time taken with exposure settings manually selected, 4) same image as in 3, this time showing the results of the One-Stage classifier - a successful detection.

Dynamic Range and Exposure Control

While the problems listed until now caused misclassifications, there is only a limited amount of what can be done as users will be using glasses and some will have long hair. We already identified the lens of a camera as being a potential issue at close distances unless it can capture a sufficiently wide angle view.

However, a more serious issue is the limited dynamic range of the camera sensor. The dynamic range is the interval between the minimum and maximum light intensity that the camera can sense. The limited dynamic range of today's consumer sensors means that high contrast scenes cannot be completely accurately captured. Due to the fact that the exposure algorithms of the camera tried to balance the scene, the faces of the participants often became completely black and lost all distinguishing features. For example, the far left picture in figure 4.11 shows a frame from an outdoors segment in Evaluation 2. As is evident in the figure, the participant's face is captured without any distinguishing features. This makes automatic eye detection impossible. This is a hardware problem that should be lessened as consumer-grade camera technologies continue to improve and increase their dynamic range. However, there are practical limits to how much dynamic range can be increased.

Moreover, given the limitations of current consumer-grade camera hardware it becomes crucial to be able to programmatically set correct exposure settings for the camera. This is because if even a single correctly exposed frame can be found it is possible to accurately detect the user's eyes. Using such past recognition results, "an area of interest" that surrounds the known prior locations of the eyes can be defined. If mobile phones and desktop computers enabled programmatic changes of exposure settings the camera could be set to focus and meter around this "area of interest". This would mitigate the issue with limited dynamic range. To demonstrate this, the second picture from the left in figure 4.11 shows the resulting frame produced by a Logitech C910 camera's automated exposure algorithms. The third picture from the left in figure 4.11 shows a frame produced by manual correction of exposure. This latter frame captures the face with enough detail for computer vision algorithms to successfully detect the eyes. To verify this, both frames were tested (with incorrect and correct exposure settings respectively) using the ONE-STAGE classifier. The rightmost picture in figure 4.11 shows that the system correctly identified the user's eyes when exposure

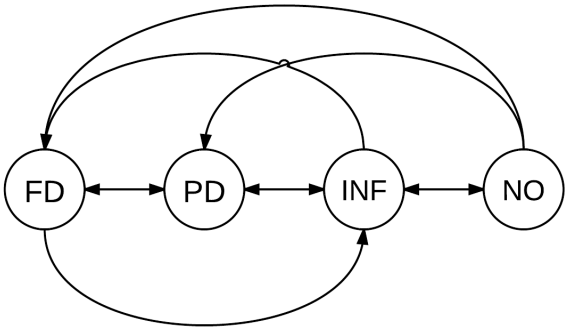


Figure 4.12: A state flow diagram of the multi-level feedback system. FD stands for Full Detections, PD for Partial Detections, INF for Inference, and NO for No Data.

settings were corrected (the eyes are indicated with a blue rectangle). However, the second picture from the left in figure 4.11 did not retain enough detail of the face for the system to detect the eyes. Interestingly, programmatic control of exposure settings is already part of a specification for USB video class devices. Unfortunately, many manufactures do not implement this specification, or they only implement a limited subset.

System State Indicator

Evaluation 2, described in Section 4.2.4, revealed that a person's distance from a camera sensor cannot always be reliably estimated in heterogeneous environments with wildly varying lighting conditions. There are likely also other edge-cases that can affect the accuracy and reliability of the distance estimator. As a way to lessen the impact of this variable detection quality, a recognition indicator is introduced. The indicator provides information to users if their eyes cannot be tracked by the system and the indicator is used in some form in all the variations of the algorithms described in this chapter (as long as they use a computer vision component).

First it is important to note that the eye-pair distance estimation uses two levels of detection. In the first detection level the system identifies an eye-pair in the camera image as a rectangular block. This detection level is sufficient, but not ideal, for estimating distance. To improve the situation, there is a second detection level. This identifies the individual eyes within the detected rectangular eye-pair region. If the eyes are detected at this level, it is possible to use their pupils as a basis for estimating the distance of the person to the camera. Knowledge of the positions of the pupils increases accuracy of the distance estimation.

Figure 4.12 shows a state flow diagram of the proposed indicator. The first state is Full Detection (FD). It means that the algorithm was able to detect an eye-pair and confirm the presence of the eyes in each subregion. Therefore, the system is confident in both the correctness of detection of the eye-pair and in the accuracy of the distance estimation. The second state is Partial Detection (PD). This is the first fallback stage. The algorithm has detected an eye-pair but it is unable to confirm the location of both eyes. This lowers the confidence in the correctness of the detection of the eye-pair. Additionally, it is no longer possible to estimate the distance of the person from the display with high accuracy.

The third state is Inference (INF). The system enters this state when the vision algorithm fails to detect any eye-pairs in the camera stream. For a short amount of time (ca. < 1 s) it is plausible a likely position of the user can be inferred based on previous data with reasonably high accuracy. Additionally, it is possible to confirm that the user is still active based on, for example, touch and keyboard input. However, the reliability of the inference will decrease with time. The last state is

No Data (NO). The system enters this state when its confidence in being able to accurately infer the likely position of the user drops below a defined threshold.

In the above design, the amount of user feedback depends on the current detection state. In the Full Detection state the system is fully working. Since interaction with the system is expected to be subtle providing minimal or no feedback is appropriate. In the Partial Detection state, the amount of feedback depends on how the system is intended to be used. If distance from the camera/display is not used as a main interaction modality, providing minimal to no feedback may be appropriate. However, if distance is the primary interaction modality then it may be necessary to provide feedback to the user. If the state does not change within a few seconds, this is a sign of a calibration issue. After a set timeout within this state, the system can inform the user to adjust their device to try to improve detection. In the implementation of the indicator used in this thesis, the lowered confidence in detections is indicated by using an orange coloured icon in the system tray.

In the Inference state the system should provide continuous feedback. For example, it can display a confidence bar that changes its size and colour depending on the confidence in its inferences. This time, however, a red icon colour may be more appropriate in order to convey that the system is not detecting the eyes at all. The system's confidence is decreasing as a function of the amount of time the system remains in this state. In the implementation of the indicator in the *DiffDisplays* system described in Chapter 7, the icon turns red as soon as the system stops being able to detect the user. In the No Data state the system cannot estimate eye-distances at all. This is indicated by a cross over the red coloured icon in the implemented version.

While this system status indicator is quite simple, it demonstrates the need for complex systems to communicate their state to the users of the system, especially in conditions where the state of the system has direct influence on its usability. Moreover, as has been already illustrated in this discussion section, there are a number of obstacles to reliable distance estimation using computer vision techniques in complex deployment conditions. Even if all the mitigation measures are successfully implemented, providing feedback about the user state is still valuable to lessen the impact of other possible edge-cases.

4.3 Tracking Using a Single RGB Camera

While the initial approach and its evaluations showed promise and helped to establish the feasibility of the computer vision approach to distance estimation, it also revealed a number of limitations. Aside from the issues covered in the previous section, the limitations with the most impact on the usefulness of this approach for running experiments were the resource intensive nature of image processing and the image resolution available. The resource limitations introduce latency into the system, while the image resolution affects both the accuracy of the distance estimations and the maximum distance that can be detected. Additionally, with increases in image resolution, the resources required to process each image increase quadratically.

This section focuses on two goals. Firstly, it codifies the algorithms used for distance estimation and coarse orientation detection in other parts of this thesis. Secondly, it describes the improvements to the algorithms, which lead to significant performance increases that allowed the system to run at 24+ frames per second even while using 5-megapixel images (2592x1944), addressing both the performance and image resolution bottlenecks.

The versions of the algorithm described in this section are primarily used for distance estimation at distances from approximately 40 cm to 300 cm. An alternative use is as a binary estimator of whether a person is facing in the direction of the camera (based on the presence or absence of detections). This can be used as a proxy for head orientation in specific display and camera setups such as the one in Chapter 7.

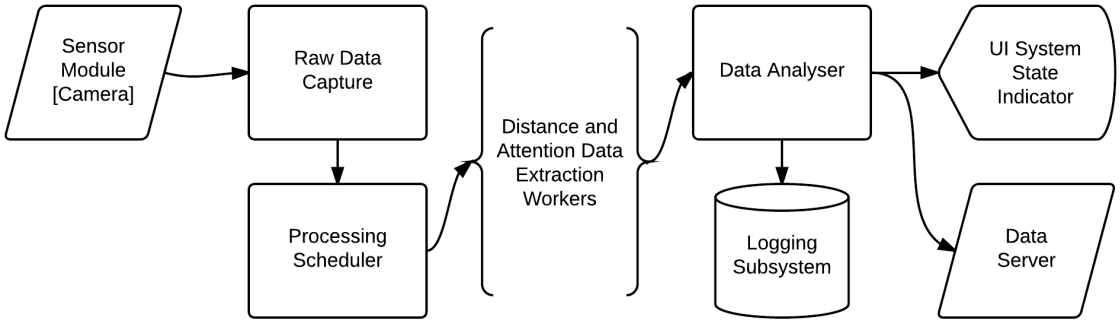


Figure 4.13: An overview of the architecture of a single node.

4.3.1 General Architecture

From the first prototype implementation, one of the main design goals has been modularity. Since the system is intended to be used as middleware or as an enabler for other applications, it is important to create a design that would be easy to integrate and highly portable. Another design consideration is the cost of deployment. The system is designed to use off-the-shelf consumer level hardware components to keep the potential cost down even if it leads to some limitations.

Modularity, Scalability and Portability

The focus on modularity, scalability and portability is clearly visible in the design of the system. As the system can consist of multiple sensors and processing nodes, every single part of the system is as loosely coupled as possible. The design centres around a *node* as the smallest unit. This allows integration of future sensor technologies with relative ease as they can be either integrated into more complex multi-sensor *nodes* or deployed as independent simple single-sensor *nodes*. Additionally, it enables the choice of any scaling of the system that is desired as one can simply deploy as many nodes as required (the limiting factor being available sensors). Moreover, the computationally intensive parts of the system are highly parallelised and the number of processing threads can be set to saturate the available compute resources or to be highly limited in scenarios where minimising the use of compute resources is the priority. Lastly, the code is based around broadly used open-source frameworks (pthreads, OpenCV) and therefore allows portability to computing platforms other than Windows. This approach also offers the possibility to be further optimised by using GPGPU processing methods.

Ease of Integration

A node is designed to be completely independent. This means that it also has an independent communication platform that can be used for integration with applications. Each node comes with a server interface, which lets applications establish a network connection, allowing two way data transmission. A node can be configured either via a local file at startup or over the network connection. Live data from the node is transmitted over the network and can also be stored locally in logs for off-line processing. This means that integration and deployment are trivial, only requiring a network connection. This design also has the advantage that the application does not need to run on the same operating system or even the same physical machine as the sensor node.

Node Design Description

Figure 4.13 shows an overview of the design of a single sensor node. The design can be divided into four conceptual parts - data capture, data processing, data analysis and output. The analyser and output parts are independent, allowing for integration of other and/or multiple sensors in the same node. Figure 4.13, illustrates the architecture of a node using the example of a relatively simple, single sensor node using a digital web camera as a sensor. The *raw data capture* module samples data from the sensor at a specified rate, which is usually the maximum allowed frame rate for the camera resolution. However, the sampling rate can be controlled internally to prevent issues such as memory bandwidth saturation. The *processing scheduler* packages the data into discrete units (usually a single frame) with accompanying structures to deal with problems related to parallel computing. The algorithms described in the following section are used within the *workers* to extract all available information, which is then passed onto the *data analyser*. This is where most of the evaluation and fusion logic resides, where the confidence in the results is established and where the output methods are decided. The resulting data can then be *logged* locally, sent to the network *server* or displayed as an *indicator* in the system tray. The indicator is based on the system state indicator proposed in Section 4.2.6. The *indicator* has three coloured states, green implies that the system is stable and that it has a high confidence in the detections, orange alerts that the confidence in the detections is decreasing and red implies that the node is unable to make any detections.

4.3.2 Deployment Details

Each node consists of a single executable with a number of support files. As mentioned above, configuration is primarily performed through a local configuration file. While the default configuration will allow the node to function, in order to achieve the highest possible accuracy, a number of parameters needs to be defined. Most of the parameters are related to the distance calculation. Specifically, the system needs to be aware of the pupil distance of the person to be tracked, the field of view of the camera and the desired camera resolution. Moreover, the spatial relationship between the camera and an associated display (if any) needs to be defined. Additionally, the number of data extraction *workers* can be specified in order to define the level to which the CPU can be taxed (one worker fully utilises a single CPU core). If the node is a complex node (using multiple sensors) additional information may be required, such as the spatial relationship between the sensors.

Information Available to Controlling Application

There is a broad range of information that is available for use by the controlling application. Apart from statistical and configuration information (e.g. processing speed in fps, calculation parameters), the application has access to the following categories of information:

- Distance - distance from the camera/sensor in millimetres, confidence in the accuracy of the distance information
- Orientation - the location of the point of view of the person being tracked within the camera field of view, from which the orientation of the camera/sensor towards the person can be derived
- Detection Accuracy - the confidence of the system in the detections and predictions it is performing
- General System State - functionality of the sensors, system state, etc.

Additional information can be extracted from raw data, which can also be provided.

4.3.3 Implementation of the Initial Approach

The next sections of this chapter describe the design and implementation of the algorithms that provide accuracy and performance beyond what the underlying feature classifiers are capable of. First, an implementation of the initial approach (outlined in Section 4.2.2) is described to establish the general detection workflow. Then, the enhancements and modifications responsible for a significant performance improvement are shown. Lastly, the results of the performance and accuracy evaluation are presented.

Feature Classifiers

The algorithm uses feature classifiers based on the Viola-Jones feature detection algorithm [VJ04], a computer vision algorithm that is readily available in the open source OpenCV library. Three specific classifiers are used. These classifiers have been shown to achieve high accuracy [CSDS+08]. Where referring to feature classifiers in this thesis, we use the same naming as Castrillon-Santana et al. [CSDS+08]. The first classifier (Ep1) detects an eye-pair. The other two classifiers (Le and Re) are for the left and right eye respectively. The reason for using three different classifiers is that while the classifiers are very accurate in controlled conditions, they tend to produce significant numbers of false positives when used alone in uncontrolled lighting conditions (see Section 4.2 or [CSDS+08] for more detailed accuracy numbers). Performing feature verification by using multiple classifiers trained on different facial features at different stages of the algorithm increases the confidence in the accuracy of detections.

Subroutine 1 Initial Approach Algorithm, *sa* is the search area. The labels to the right point to details of named subroutines or point out where specific feature classifiers are used.

```
1: sa ← image
2: eye-pairs ← OpenCVscan[sa, Ep1]                          ▷ Ep1 classifier
3: if length[eye-pairs] = 0 then
4:     fail and take no action
5: else
6:     for all eye-pairs do
7:         verification(eye-pair)                            ▷ Subroutine 2
8:     end for
9:     return selectClosestCandidate(eye-pairs)              ▷ Subroutine 3
10: end if
```

Subroutine 2 verification(*eye-pair*)

```
1: rectangles right, left
2: centre[left] ← (centreY[eye-pair], centreX[eye-pair] + width[eye-pair]/4)
3: centre[right] ← (centreY[eye-pair], centreX[eye-pair] − width[eye-pair]/4)
4: height[left] ← height[right] ← 1.2·height[eye-pair]
5: width[left] ← width[right] ← 1.6 · (width[eye-pair]/2)
6: l-eye ← selectLargest[OpenCVscan[left, Le]]              ▷ Le classifier
7: r-eye ← selectLargest[OpenCVscan[right, Re]]             ▷ Re classifier
8: return eye-pair, l-eye, r-eye
```

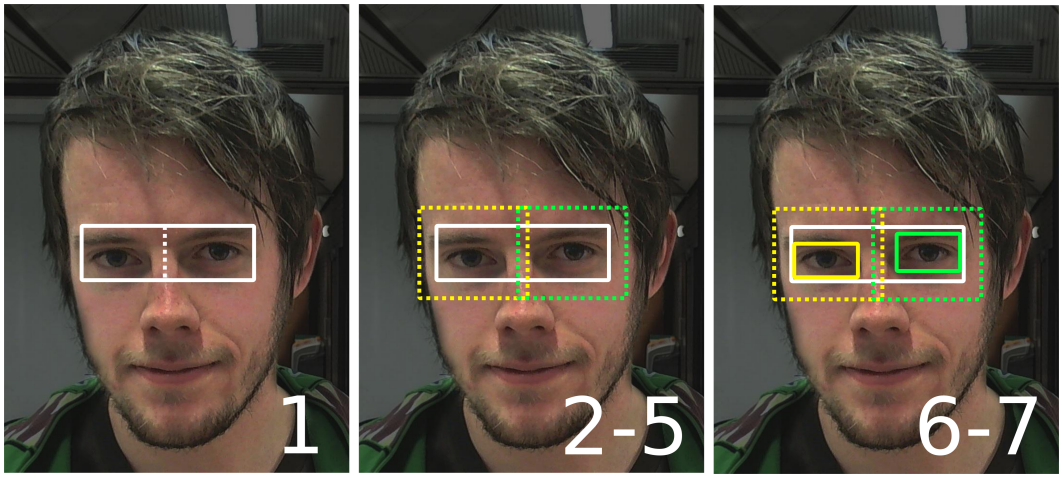


Figure 4.14: A visual representation of the different steps of the verification subroutine. Numbers correspond to lines in Subroutine 2.

The Initial Approach Algorithm

The general flow of the algorithm is shown in Subroutine 1 and described here in more detail. The algorithm can be divided into five conceptual parts - search area definition, eye-pair search, individual eye search, candidate selection, and presentation of results.

In the first part of the algorithm, an eye-pair search area is established. For the benchmark algorithm, this is the entire image frame sent by the camera. Using the entire image, there is a guarantee that even if the user changes their position significantly between frames, they will be found as long as they stay within the field of view of the camera.

Secondly, a search for eye-pairs within the defined area is performed. This search is a straightforward search using the feature detection algorithm described earlier, using the eye-pair classifier. In order to speed up the search, 1.2 scaling factor (this scaling factor defines how much the image will be scaled for each iteration of the feature detection algorithm) and canny pruning are used.

Thirdly, for each eye-pair found, a more detailed search for each of the individual eyes (within a partitioned area somewhat larger than the eye-pair itself) is carried out. This is captured in Subroutine 2. Since the eye-pair classifier returns the bounding rectangle for the eye-pair as a whole, it does not necessarily contain much of the area surrounding the eyes (see the leftmost image in Figure 4.14 for illustration). The bounding rectangle often ends with the outer corner of the eyes and when the user's head is partially rotated, the eyelids can be very close to the rectangle boundaries. This can make detection of single eyes just within this area unreliable. In order to counter this effect, after the eye-pair is divided into the left and right sub-areas, their area is increased by 60% vertically and 20% horizontally.

Fourthly, the best candidate eye-pair is selected based on a predefined condition. The candidate selection part of the algorithm is described in detail in Subroutine 3. Essentially, the fitness of the candidates is judged on two characteristics - the validated eyes and the width of the containing rectangle of the eye-pair. If both of the eyes were confirmed in the third step, a candidate is better if the size of the containing rectangles for the individual eyes is closer than that of the current primary candidate. For cases where no eye-pairs with verified individual eyes exist, the secondary metric is the width of the containing rectangle of the eye-pair. This means that the secondary candidate will always also be the eye-pair that is closest to the camera.

Lastly, all the information about the selected eye-pair is returned together with the type of detection it is. There are three categories of detections. Firstly, there is *no detection*, which is used when no eye-pairs are found. A *full detection* is defined as a detected eye-pair with both eyes validated. Information about the best primary candidate is returned on *full detection*. A *partial detection* is defined as a detected eye-pair, which, does not have both eyes validated. The confidence in this type of detection is inherently lower due to a higher chance of false positives. Information about the best secondary candidate is used with this type of detection.

Subroutine 3 selectClosestCandidate(*eye-pairs*)

```
1: for all eye-pairs do
2:     if eye-pair has both eyes detected and the sizes of the rectangles containing each eye match
   each other more closely than current primary candidate then
3:         primary candidate ← eye-pair
4:     else if eye-pair does not have both eyes detected but width[eye-pair] > width[secondary candidate]
   then
5:         secondary candidate ← eye-pair
6:     end if
7: end for
8: if primary candidate ≠ Nil then
9:     return primary candidate                                          ▷ Full Detection
10: else
11:    return secondary candidate                                        ▷ Partial Detection
12: end if
```

4.3.4 Improvements

While the initial algorithm implementation described in the previous section is accurate and provides a flexible measure of confidence of its results, it has a significant disadvantage, which is directly related to the nature of accuracy of feature detection using image processing with Viola-Jones feature classifier. In order for a feature to be detected it needs to occupy a certain minimum area within the image that contains enough detail for the feature to be distinguished. As an example, the EP1 eye-pair classifier can only detect eye-pairs larger than 45×11 pixels[CSDS+08]. This is the minimum amount of detail needed, which means that simply interpolating a smaller image would not help as it would not contain sufficient information.

This effect puts a limit on the maximum distance that an algorithm using these feature classifiers can detect. For example, for a person with 6.5 cm pupil distance, a camera with a 60° field of view and a VGA resolution (640×480) cannot reliably detect distances greater than approx. 150 cm using the EP1 classifier. Additionally, the granularity of the distance detection is also affected because as a person moves further from the camera, the pixel distance between their pupils is decreasing logarithmically, which translates to exponentially decreasing spatial resolution. This means that in order to increase the maximum detection distance as well as the accuracy of detections, it is necessary to use images with resolution that is as high as possible. The disadvantage of this approach is that the amount of pixels that need to be processed increases quadratically with image size. As will be reported in the performance evaluation in Section 4.3.5, processing a single 5 megapixel image takes almost 1.5 seconds. This significantly limits practical use of the algorithm due to both the 1.5 second latency and also the implied 0.66 frames per second processing framerate. Therefore, the next step in the development of the algorithm is to increase its speed of execution, while maintaining its accuracy.

Since both the processing power and the amount of data to process[4] are a bottleneck, the problem can be approached from two directions. Firstly, one can increase the processing power. The algorithm will fill an entire CPU core automatically. In order to increase the processing power, the code is parallelised so that any number of worker threads can be used. This allows maximisation of the CPU usage no matter how powerful the CPU is or how many cores it has. However, since the systems using this algorithm are expected to be used in scenarios with an active computer user it is not a very good solution by itself. This is because utilising the entire CPU adversely affects the responsiveness of a computer when it is used for other tasks simultaneously.

Additionally, while parallel processing increases total throughput almost linearly (in this case), the processing time per frame does not change. The result is that while the distance is updated at a higher frequency, the reported distance is still the same 1.5 seconds in the past. So, while the parallelisation is useful, it is necessary to look at decreasing the amount of data for processing for a more significant improvement in speed.

The next two sections summarise the improvements to the initial implementation of the algorithm. Specifically, the first stage of the algorithm (where the initial search for eye-pairs is performed) and the fourth stage (where the best matching candidate eye-pair is selected) are changed. Subroutine 4 shows the outline of the algorithm with improvements.

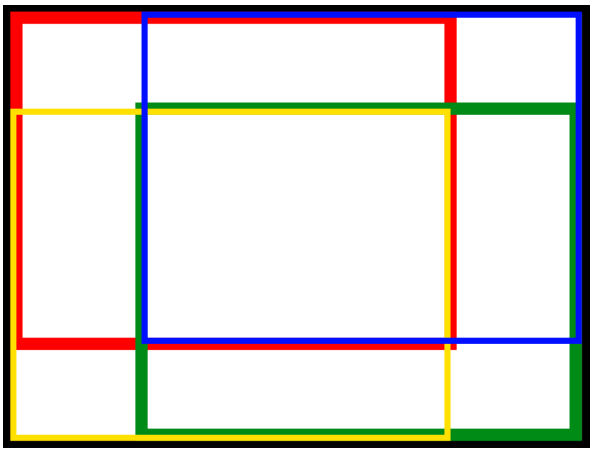


Figure 4.15: A visual representation of the four possible corner alignments within the camera image as per line 30 in Subroutine 5.

Subroutine 4 Improved Algorithm

```
1: search area ← searchAreaPrediction()                    ▷ Subroutine 5
2: eye-pairs ← OpenCVscan[sa, Ep1]                         ▷ Ep1 classifier
3: if length[eye-pairs] = 0 then
4:     fail and take no action
5: else
6:     return selectSimilarCandidate(eye-pairs)            ▷ Subroutine 6
7: end if
```

[4] which are effectively two sides of the same coin

Tracking and Dynamic Search Area Allocation

The first area of improvement is in the first part of the initial approach algorithm, namely the search area in which the algorithm looks for eye-pairs. The position of and eye-pair in two subsequent frames in an image stream is similar. This similarity increases as the time difference between the two frames decreases. This leads to two conclusions. By tracking the position of the last detected eye-pair, it should be possible to decrease the size of the search area within an image. Additionally, the accuracy of the estimated position within the image will increase as the processing rate increases because the tracked person will have less time to move between frames.

The search area in the improved algorithm is determined based on the type and recency of tracking information available. A detailed description of this can be found in Subroutine 5. If a detection was made in the last frame, the search area is tightly restricted to the position where the eye-pair is predicted to be. This position is based on the movement vector between the last two detections. The size of the search area depends on the confidence in the accuracy of the last detection. With the exception of the low confidence detection, the search area takes into account the size of the previously detected eye-pair as well as the movement vector (this is to compensate for potential changes in speed of movement). The low confidence variant simply searches an area equal to half the height and width of the camera image (effectively 25% of the area). If there was no detection in the last frame, the size of the area is determined based on the recency of the last detection. If a detection was made less than two frames ago (not counting the current frame), the area equal to 25% of the size of the camera image (width/2×height/2) is searched, positioned so the last known detection is in the centre of the search area. If a detection was made exactly two frames ago, the search area size is increased to 56.25% of the camera image area (width*0.75×height*0.75). If the last detection was earlier than that, the same 56.25% area is used but it is aligned with one of the corners (a different corner in each iteration) of the camera image. This means that the algorithm will search only a little more than half the area of the image, but every single pixel of the camera image will be searched at least once in four frames (and even this applies only to a small corner area). Additionally, the areas of the image where a tracked person is more likely to be are searched more frequently, e.g. the central image region is searched in every frame. See Figure 4.15 for visual illustration.

Candidate Selection

Once the algorithm has found a number of eye-pairs in the image, it is necessary to select one or more of them as the closes match(es) to the tracked eye-pair from the frame before. The improved candidate selection process is based on a similarity metric. The similarity metric consists of the inverse of three components, horizontal dissimilarity, vertical dissimilarity and area dissimilarity. They are all based on the relationship between the last detection and the current possible detection (eye-pair being examined). The components are computed as follows:

$$\begin{aligned} S &= 1 - (0.4 \cdot H + 0.4 \cdot V + 0.2 \cdot A) \\ H &= \frac{|x_c - x_l|}{w_c + w_l} \\ V &= \frac{|y_c - y_l|}{2 \cdot (h_c + h_l)} \\ A &= \frac{|A_l - A_c|}{A_l} \end{aligned} \tag{4.2}$$

where S is the similarity, H, V and A are the horizontal, vertical and area dissimilarity, respectively, x and y refer to the coordinates of the centre of an eye-pair, w and h to the width and height of the

Subroutine 5 searchAreaPrediction(), *sa* is the search area, *ld* is the last detection

```
1: ld ← lastDetection()
2: if ld ≠ Nil ∧ recency[ld] < 1 then
3:     hm ← horizontalMovement[ld]
4:     vm ← verticalMovement[ld]
5:     if detectionType[ld] = full then
6:         width[sa] ← 2·width[ld]+4·hm
7:         height[sa] ← 3·height[ld]+4·vm
8:         centre[sa] ← centre[ld] offset by hm and vm
9:     else if detectionType[ld] = partial & confidence[ld] > 0.25 then
10:        width[sa] ← 3·width[ld]+(4·hm)/confidence[ld]
11:        height[sa] ← 3·height[ld]+(4·vm)/confidence[ld]
12:        centre[sa] ← centre[ld] offset by hm and vm
13:    else
14:        width[sa] ← 0.5·width[image]
15:        height[sa] ← 0.5·height[image]
16:        centre[sa] ← centre[ld]
17:    end if
18: else if ld ≠ Nil ∧ recency[ld] ≥ 1 then
19:    if recency[ld]< 2 then
20:        width[sa] ← 0.5·width[image]
21:        height[sa] ← 0.5·height[image]
22:        centre[sa] ← centre[ld]
23:    else if recency[ld]= 2 then
24:        width[sa] ← 0.75·width[image]
25:        height[sa] ← 0.75·height[image]
26:        centre[sa] ← centre[ld]
27:    else
28:        width[sa] ← 0.75·width[image]
29:        height[sa] ← 0.75·height[image]
30:        sa aligned with the (recency[ld] mod4)-th corner of image      ▷ Figure 4.15
31:    end if
32: else
33:    sa ← image
34: end if
35: if ld ≠ Nil ∧ recency[ld] < 1∧ width[ld] > 100 then
36:    scale sa by factor width[ld]/100
37: end if
38: return sa
```

bounding rectangle of the eye-pair and subscripts c and l refer to the current and last detection, respectively.

The selection process described in Subroutine 6 has a number of advantages. It helps increase the accuracy of the selection process by favouring candidates that are close to the last successful detection in size and position. This helps avoid false positives as the position of the tracked person's eyes within the image is likely to be close to its last position. Moreover, by favouring similar size results, the distance computed from the eye-pair will be more consistent between frames.

Subroutine 6 selectSimilarCandidate(*eye-pairs*), *ld* refers to the last detection

```
1: if ld ≠ Nil then
2:     for all eye-pairs do
3:         similarity[eye-pair] ← S(eye-pair, ld)                          ▷ Eq. 4.2
4:         if similarity[eye-pair] < similarity[primary candidate] ∨ best candidate = Nil then
5:             verification(eye-pair)                                      ▷ Subroutine 2
6:             if both eyes found then
7:                 primary candidate ← eye-pair
8:             else if similarity[eye-pair] > similarity[secondary candidate] then
9:                 secondary candidate ← eye-pair
10:            end if
11:        end if
12:    end for
13: else
14:    use selectClosestCandidate(eye-pairs) from benchmark algorithm      ▷ Subroutine 3
15: end if
16: if primary candidate ≠ Nil then
17:    return primary candidate                                            ▷ Full Detection
18: else
19:    return secondary candidate                                          ▷ Partial Detection
20: end if
```

4.3.5 Performance and Accuracy Evaluation

In order to test the effectiveness of the improved algorithm, an evaluation of both the speed as well as the accuracy of the two algorithms was conducted. Two equally sized datasets of camera images were created for testing. Both datasets were created to be indicative of a generic workspace environment. One dataset contained only positive samples while the other only contained negative samples. The images for both datasets were collected at 10 frames per second with resolution 2592×1944 pixels.

The datasets were the same size (1000 frames each), in order to balance the expected amount of True Positives and True Negatives. Additionally, since especially the amount of data processed by the improved algorithm changes dynamically, it is important to test the worst case detection scenario (no detections for significant periods of time) as well.

The positive dataset consisted of 1000 ordered frames of a single person moving towards and away from the camera with distances ranging from 35 cm to 350 cm. Sideways movement was also present. The single subject was looking at the camera at all times, but all the characteristic traits of normal use (eye blinks, motion blur, etc.) were retained in the dataset.

The negative dataset consisted of 1000 ordered frames without a single frame of a person looking at the camera. However, in order to simulate the office environment better and to make the data richer, a person doing desk work was present in the background with their back to the camera and several times a person walked through the camera's field of view without looking at the camera.

The datasets were processed by the benchmark algorithm and the improved algorithm. The speed tests were performed on a 3.2 GHz Intel Core i3 CPU using a single processing thread. Due to the fact that various processes of the operating system were running in the background, which could potentially influence the run time, the performance on each dataset was measured 10 times to gain a more accurate average value. Additionally, every processed frame was saved to the hard drive, encoded with information about the detection (if any) and a human judge confirmed whether or not the detection was correct.

Dataset	Algorithm	Accuracy	TP	TN	FP	FN
Total	Benchmark	98.75%	49.85%	48.90%	1.15%	0.10%
	Improved	99.00%	50.05%	48.95%	1.00%	0%
Positive	Benchmark	99.70%	99.70%	0%	0.10%	0.20%
	Improved	100%	100%	0%	0%	0%
Negative	Benchmark	97.80%	0%	97.80%	2.20%	0%
	Improved	98.00%	0.10%	97.90%	2.00%	0%

Table 4.4: Comparison of accuracy of algorithms. The accuracy breakdown is also included, showing True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).

Accuracy

Table 4.4 shows the accuracy of the two algorithms. The algorithms performed very well, showing very high accuracy for both datasets. While the data in the table is mostly self explanatory, there are several points of interest. Firstly, the improved algorithm made one true positive detection in the negative dataset. In order to make sure that the there would be no positive detection, the person passing through the camera's field of view wore sun glasses to obscure the eyes. However, in this particular frame, the improved algorithm still managed to detect the eyes, so the detection had to be acknowledged. However, it was only a partial detection, which means it was a lower confidence detection.

Another item of note is that of the true positive detections, on average 6.5% were partial (lower confidence) detections for both algorithms. The benchmark algorithm made one misclassification, where the algorithm made a full (high-confidence) detection, which was incorrect. Although the person in the image was looking at the camera at the time, the detection was made in an area around their lips. Therefore, this misclassification was included as a false positive. All the other false positives, for both algorithms, were partial (low-confidence) detections only, and they were mostly caused by shapes within the image with a similar structure to an eye-pair. The false negatives for the benchmark algorithm were due to the tracked person blinking, in both cases.

The results above confirm that the improved algorithm maintained the accuracy of detections. With the first goal achieved, it remains to be seen if and by how much the processing speed increased with the improved algorithm.

Processing Speed

Figure 4.16 shows the mean processing time per frame for the positive and negative datasets. It is clearly visible that the processing speed of the benchmark algorithm was very consistent in both datasets, due to the fact that most of the time was spent in the early parts of the algorithm — searching for eye-pairs. The reason why the benchmark algorithm was slightly faster in the negative dataset compared to the positive dataset is due to the latter parts of the algorithm being skipped over when no eye-pairs were found. This also reaffirms the earlier observation that the search area for the initial stages of the algorithm is a major performance bottleneck.

The improved algorithm showed a marked boost in processing speed. In the negative dataset, where due to lack of tracking data the algorithm was forced to use the corner-aligned large search area, the improved algorithm used only $\approx$55% of the time needed by the benchmark algorithm. This is consistent with only 56.25% of the image area being searched in each frame. The performance enhancement was even more noticeable for the positive dataset. There, the improved algorithm managed to process a single frame almost 37 times faster than the benchmark algorithm. This

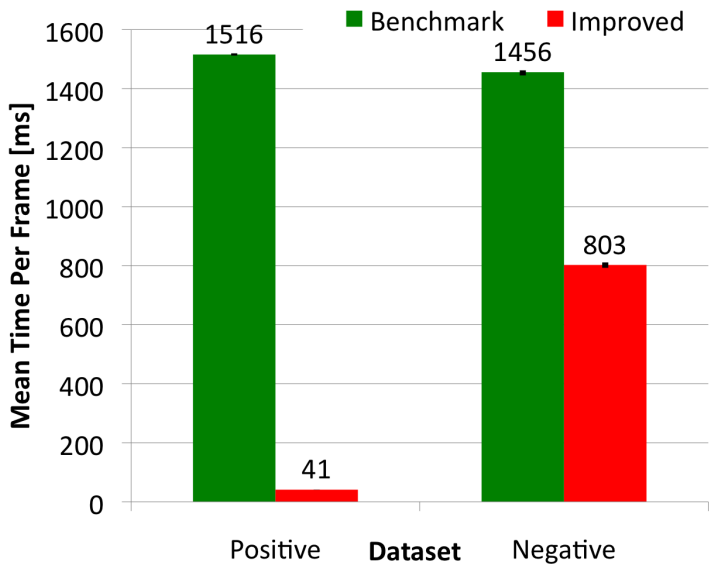


Figure 4.16: The mean processing time per frame of the two algorithms for each of the two datasets.

marked increase in processing speed was mostly due to the improvements in eye-pair search area predictions using historical tracking data.

The collected data shows that the improved algorithm is efficient enough to process 2592×1944 pixel images at ≈24 fps when tracking data for a person is available. The actual processing framerate will vary with the distance at which people are tracked as the closer the tracked person is, the more of the image area they will occupy (and by extension the slower the processing will be). The positive dataset contained frames of people at the entire range of distances in approximately equal proportion, so the processing time is representative for a deployment with a wide range of tracking distances. For more limited tracking distances, selecting the lowest possible resolution to capture the required distance will provide an automatic performance boost.

In order to make the performance even more consistent in deployments with an expectation of variable interaction distances, an after-the-fact improvement to the algorithm was made. This modification results in an optional scaling of the image used as the search area if the width of the image is greater than 100 pixels (the change is reflected on lines 35 to 37 in Subroutine 5). This significantly improves the worst case processing speed at close distances when very recent (and thus highly reliable) historical tracking data is available. It does not affect performance at larger distances due to the 100 pixel width limit.

When a person is no longer tracked, the improved algorithm processes images at a rate of ≈1.25 fps (for a 5-megapixel image). In practical use with people present in the images, this translates to a ≈24 fps performance with a ≈800 ms acquisition lag. Importantly, these performance figures are for a single thread and since the system scales almost linearly (for a reasonably small number of cores). And again, as with the active tracking performance, selecting the lowest resolution required for the deployment will provide a further performance boost.

The last notable observation is that with the improvements to the processing speed, the camera hardware becomes the performance bottleneck as the Logitech C910 camera cannot output 2592x1944 images at a faster rate than ≈10 fps meaning that the image processing algorithm is unlikely to be a bottleneck without a large increase in image resolution and framerate.

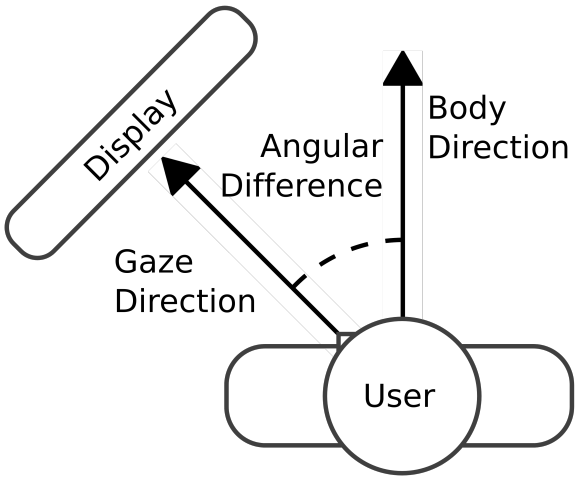


Figure 4.17: A diagram illustrating how angular difference was measured in the evaluation of the head orientation detector.

The improved algorithm has been used in three prototypes in this thesis. The first two prototypes (*Multi-View Train Board* and *Multi-View Video Player*) are described in Chapter 5. The third prototype, *DiffDisplays*, utilised a modified version of the algorithm, which is described in the next section.

4.3.6 Additional Evaluation for Use as an Orientation Detector

DiffDisplays, described in detail in Chapter 7, relies on the ability to tell whether a person is looking in the direction of a display or not in order to trigger one of four visualisation techniques. In order to support this type of interactions, the improved version of the distance detection algorithm was used as a basis for a display-level orientation detector using commodity hardware. The system tracks users' head orientation using web cameras mounted on displays. The head orientation is used as a proxy for visual focus. While the system does not use the distance of a person from the display as was the focus up to now, the detections and their quality are used to determine whether the person is looking at a specific display or not. If a person is looking at a display, the algorithm should be able to detect the person's eyes in the camera image. If the eyes cannot be found, the person is likely not looking at the display. To ensure the orientation detector is sufficiently accurate and robust to be deployed, an evaluation with seven participants was carried out.

Method

The system was deployed to three dual-core iMac computers with 20-inch displays (1680×1050 pixel resolution). A Logitech C910 web camera was attached to each display. All computers were running the Windows 7 operating system. Each display was positioned at 76.5 cm from the user's default sitting position. This distance was based on the results of a study of comfortable viewing distances for computer displays, which reported this distance as the most comfortable viewing distance on average [GHN84].

Six angular differences were tested (15°, 30°, 45°, 60°, 75° and 90°). Figure 4.17 illustrates how angular difference was defined in the evaluation. The six different angles were chosen to sample the whole visual field of the participants. Seven university students were recruited as participants (6 male). Their ages ranged between 19 and 26. Their skin colour and hairstyle varied but none of them wore glasses or used other reading aids.

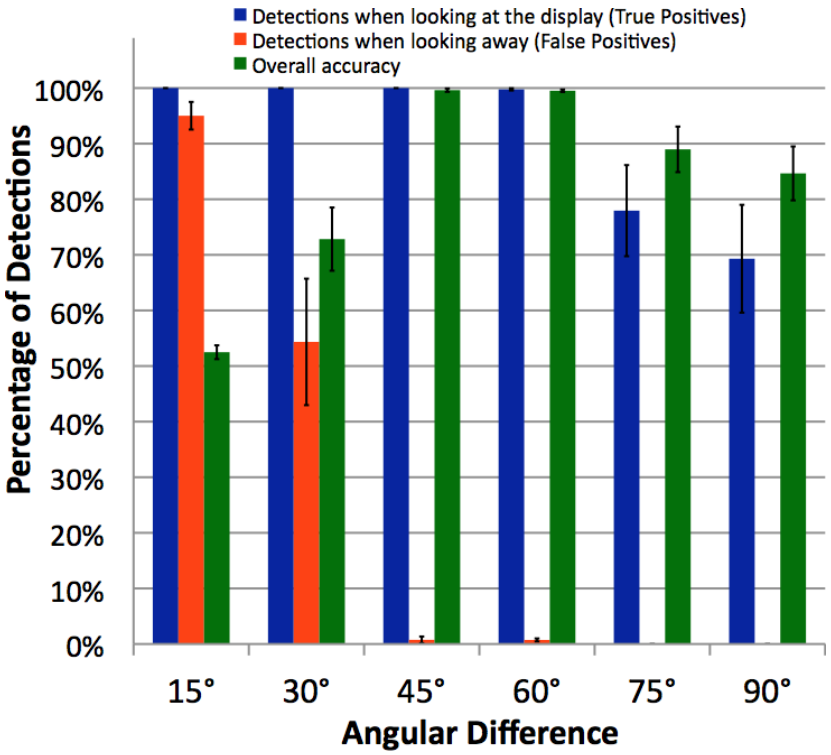


Figure 4.18: This figure shows the results of a study of accuracy of detection at different angular differences. The first column (in blue) shows the percentage of True Positives. The second column (in red) shows the percentage False Positives. The third column (in green) shows the overall accuracy. The error bars show standard error of the data.

Each participant was asked to follow a sequence of spoken instructions directing them to look at a centre target (centre of a display) and side targets (centres of other displays) in turn. The participants were also instructed to keep their body facing the centre target and to only move their head and/or their eyes. As body position is not important during the actual detection, the goal was for the participants to only move their head to exaggerate any possible effect imposed by the eye and head movement.[5] They were asked to remain at the specified distance from the displays to maintain consistent viewing angles. For each examined angle, every participant spent approximately 25 seconds looking at the centre target and 20 seconds looking at each side target. This generated approximately seven minutes of video data per participant. The orientation detector then detected the participant's eyes for each video frame. Thereafter, the accuracy of the orientation detector was assessed by visually comparing the algorithm's detection results against the ground truth in each image.

Results

Figure 4.18 shows the percentage of video frames with successful detections by the detection algorithm. The blue bars (first column) show the number of detections when the user is looking the centre of the display (true positives). The red bars (middle column) show the detections while

[5]The Appendix provides further considerations related to human vision, which should be taken into account by designers of systems like this.

the user is looking away from the display at a specific angle (false positives). The green bars (third column) show the overall accuracy of the algorithm.

For angular differences below 45°, even though the visual focus detector performs perfectly in terms of true positives, the overall accuracy is significantly lowered due to the high number of false positives when the user is looking away. However, the amount of false positives decreases as the angular difference increases, and at a 45° angular difference the overall accuracy is higher than 98%.

For extreme angles greater than 60°, detection accuracy decreases substantially. In these cases, the participants tended to not move their head much, but instead moved their pupils all the way into the corner of their eyes to see the display. This was expected as the participants were specifically instructed to not move their body, which is what would naturally occur. This meant that for much of the time the participants' eyes appeared very different from how they typically appear when they look straight at a display.

While 20-inch displays were used in this evaluation the determinant is the angular difference. The results show that the modified algorithm can be used as a display-level orientation detector. A 45° angular difference provided the best performance. As a calibration point, two 27-inch displays, positioned at a comfortable viewing distance 76.5 cm away from a person bezel-to-bezel, have an angular difference close to 45°. In fact, taking advantage of this observation, two 27-inch displays were used in the setup for the case study described in Chapter 7.

4.4 Tracking Using an RGB Camera and a Depth Camera

Being able to track distance and even coarse grained orientation using a single RGB camera proved a potent enabler for running a number of case studies. However, while the algorithm could be modified for multi-person scenarios, it was not well designed for that purpose, mainly because it lacked any notion of user identity. This section introduces an extensively modified version of the most efficient version of the single RGB algorithm. It uses a depth sensing camera (in this case a KinectV1) *in combination* with previously described computer vision algorithms to detect the user's eyes in regular RGB camera streams. The tracking system consists of four separate parts: user identification, head position tracking, attention detection, distance estimation and distance estimate correction.

The implementation uses the OpenCV and OpenNI frameworks coupled with custom code. OpenCV offers implementations of standard computer vision algorithms as well as access to camera hardware. The OpenNI framework enables programmatic interactions with the KinectV1 depth sensor and its data.

This algorithm was designed to provide multi-person distance and position tracking, while retaining the coarse grained estimation of whether a tracked person is looking in the direction of the camera. The expected range of distances for tracking is approximately from 40 cm to 500 cm. The tracked volume is further constrained by the field of view of the cameras (approx. 60 degrees horizontally).

4.4.1 User Identification

To track multiple users, it is essential to have a robust user identification mechanism. The KinectV1's depth-based blob segmentation accessible from OpenNI is used within our system as it has proved more reliable and less resource intensive compared to a computer vision approach. Due to the use of depth data, this approach is relatively robust to body occlusion and fast movement. However, this approach may lead to misidentification of users when they leave the field of view and later rejoin at a different distance. The user identification was tested with up to four users.

4.4.2 Head Position Tracking

After a user has been identified, the position of their head within the depth/RGB images is established as described below. This is a necessary step that allows the system to perform multi-user tracking in real time. This is because it enables a significant reduction to the size of the search area within the images. The tracking is accomplished with a cascade of three head position predictors. The primary predictor uses the last known position of the eye-pair from the last iteration of the algorithm. If the position of the user's eye-pair is known, it is used as the centre of the search area. If the eye-pair data is not available, the secondary predictor is based on the skeleton data from the depth camera. The head joint is used as the centre of the search area. If the skeleton data is not available, then the tertiary predictor attempts to predict where the head is from the depth blob used to identify the user.

4.4.3 Head Orientation Detection

Once the search area for the likely position of the user's head has been established, the rectangular search area is translated into the coordinate space of the RGB camera and a search for the user's eyes is performed. The algorithm used to perform the search and candidate selection is the same as described in Subroutine 4, with one exception. The first step (searchAreaPrediction) is performed by the detectors described in 4.4.2, rather than those from the standard algorithm described in Subroutine 5.

To summarise this phase of the process for convenience, in the first stage of the search a classifier attempts to locate the user's eye-pair. If successful, the second stage classifiers attempt to confirm the result by locating the left and right eye separately in the left and right halves of the eye-pair area. The confidence of the attention detector depends on which of the search stages were successful. The system will report either a full detection (both first and second stage detections were successful), a partial detection (only the eye-pair was detected), or no detection. The detector will also report detailed information about the detected eyes. This is used for distance estimation, as described in the following section. It can also be used for head position tracking in the future. In addition to the above, the eye-pair data provides information about whether or not the tracked person is looking in the direction of the cameras or not. This can be used as an indicator of visual focus or attention.

4.4.4 Distance Estimation

Distance estimation is performed by a cascade of estimators that use available sensor data. All of the estimators estimate the distance from points within the depth data, the only difference is the method of choosing the sampling points. The primary estimator uses the points between the eyes translated into the coordinate space of the depth camera, if the eye-pair data from the attention detector is available. The secondary estimator uses points between the head and neck joint of the skeleton data, if it is available. The tertiary estimator uses the mean distance of the top 25% of the user's blob if only the depth-segmented user blob is available.

4.4.5 Distance Estimation Correction Model

An evaluation described later in this section revealed that the KinectV1 depth camera systematically over-estimates distances as a function of nominal distance (see Figure 4.19b). To compensate for this overestimation error, the nominal depth values are adjusted using a pre-computed linear regression model. The linear regression correction model is: $y = 0.9005x$.[6] Experimental data shows that this correction model explains 99% of the variance of the overestimation error ($R^2 = 0.99$). The final

[6]The model for figure 4.19c also includes an offset of 48.411 mm to account for the mean distance between user's feet and their eyes.

result is that when users are between 0.5 and 5 metres away from the display, the system can reliably estimate their distance from the display with a maximum error of just over 10 cm (see Figure 4.19c).

4.4.6 Tracking Latency

Using a 2.8GHz Quad-Core Intel Core i5 processor it is possible to track four users with a latency of approximately 30–40 ms at 20–30 fps. To speed up the tracking of multiple users to this level the tracking procedure is parallelised. As mentioned before, the OpenCV implementation of the Viola-Jones feature tracker is used for the computer vision parts and OpenNI is used to access the KinectV1 data. Unfortunately, OpenCV and OpenNI are difficult to multithread due to critical data structures being exposed in shared memory without appropriate locking mechanisms.

To overcome these limitations, a series of locks are used around OpenCV and OpenNI's core data structures and a separate worker thread is used for each person tracked. This enables multiple users to be tracked at approximately the same speed as a single user if there are enough available CPU cores on the machine performing the tracking.

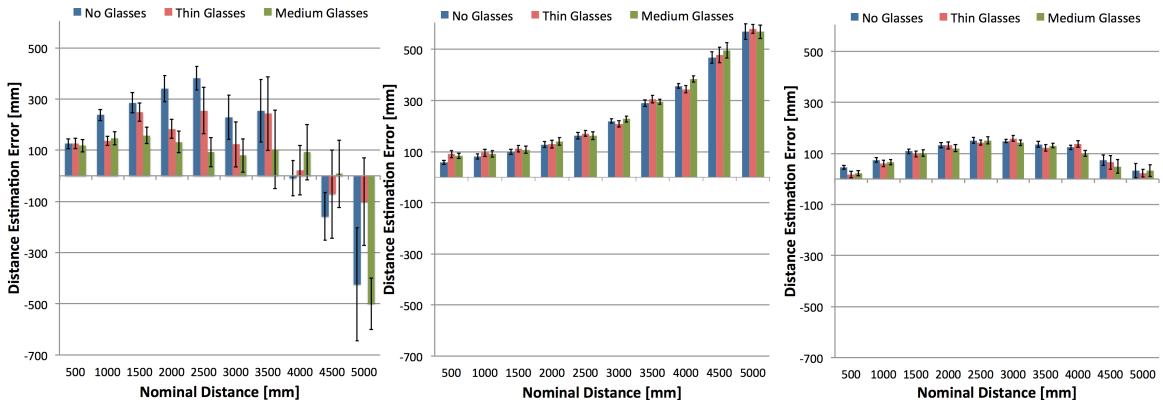
4.4.7 Evaluation

To determine the potential of fusing computer vision and depth sensing an evaluation was performed. Eight participants were recruited (three females and five males; their ages ranged from 21 to 39) from a local university campus. The experiment followed a within-subjects design with two factors: Glasses (participants wearing no glasses, participants wearing glasses with a thin frame, and participants wearing glasses with a thick frame) and Sensor (Computer Vision Only, KinectV1-CV Fusion, and KinectV1-CV Fusion Corrected). The Computer Vision Only condition used the distance of the participant's pupils that was available from the attention detector to estimate distance using a 5 megapixel (2592×1944 pixels) image taken with a Logitech C910 RGB camera (using only the Algorithm 4 described in Section 4.3.4. The KinectV1-CV Fusion condition used the fusion algorithm just described, without the pre-computed correction model. The KinectV1-CV Fusion Corrected condition used the fusion algorithm together with the correction model.

The RGB camera was positioned on top of the KinectV1 sensor. The floor was marked at 50 cm intervals at a range from 50 cm to 5 metres. Each participant was asked to stand with their feet aligned to each of the distance markers, while the study administrator manually read the distance value from each of the sensors. The process was repeated for each participant three times. Each time the participant either wore glasses with thin or thick frames, or no glasses at all (see Figure 4.20 for and image of the glasses used).

Figures 4.19a, 4.19b and 4.19c show the distance estimation error for Computer Vision Only, KinectV1-CV Fusion, and KinectV1-CV Fusion Corrected respectively. In each case, perfect performance would be represented by a constant error of approximately 5 cm (due to the difference in the position between the tips of the feet of the participants and their eyes). As is evident in the figure, the final system that uses the linear regression correction model resulted in an estimation error less than 10 cm for a range between 0.5 and 5 metres. The evaluation also showed that the system can accurately detect the user even if the user wears glasses.

The estimation profile of Computer Vision Only follows a distinctly different curve to the KinectV1. At distances closer than four metres, Computer Vision overestimates the distance, while beyond four metres, Computer Vision Only starts to severely underestimate the distance. This behaviour is consistent with the underlying algorithm, where up to four metres, the algorithm achieves high confidence detections using a combination of three different classifiers (the eye-pair, and the left and right eye). Beyond the four metre point, the algorithm can only rely on lower confidence single classifier detections (the eye-pair) because too few pixels capture the individual eyes for the single



(a) Computer Vision Only (b) KinectV1-CV Fusion (c) KinectV1-CV Fusion–Corrected

Figure 4.19: Mean distance estimation error using computer vision-only, using computer vision-guided KinectV1, and computer vision-guided KinectV1 with a pre-computed correction made using a linear regression model. The three conditions in the experiment were participants wearing no glasses, glasses with a thin rim, and glasses with a rim of medium thickness. The error bars show standard error.

Figure 4.20: An image of the glasses with highly reflective lenses used. The thin rimmed glasses are shown above the medium rimmed glasses.

eye classifiers to work. The accuracy decreases because the precise location of the eyes can no longer be established and the location is instead estimated from the bounding rectangle of the eye-pair.

The distance estimation of KinectV1-CV Fusion is very stable whether participants are wearing glasses or not. However, there is a clear increase in inaccuracy as the distance increases. While this is partially due to the decrease in spatial resolution, there seems to be a bias towards overestimation that increases with distance. As can be seen in Figure 4.19c, correcting the KinectV1-CV Fusion model using a pre-computed linear regression model substantially reduces estimation errors and results in highly accurate distance estimation.

4.4.8 Discussion

It is possible to perform fast tracking of multiple users by using the KinectV1 data or by estimating a distance directly from a blob obtained from the depth data (although this is non-trivial and body occlusion is a serious problem). However, the Computer Vision-KinectV1 fusion procedure provides three distinct advantages to just using the KinectV1. Firstly, the KinectV1 only allows simultaneous skeleton tracking of two people, whereas the algorithms presented in this section allow up to four people to be tracked simultaneously.

Secondly, it is possible to obtain a more specific and accurate distance estimation compared to what is possible using just the KinectV1 skeleton interface. The range obtainable using the fusion system is between 50 cm and 5 metres compared to the range of the KinectV1 skeleton tracking, which is available between 80 cm and 4 metres. For the fusion system, 5 metres is the maximum range tested; the actual maximum range is likely even greater. The limitation is the availability of user blobs from the OpenNI user tracker (which starts to degrade at around 4.5 m) rather than the distance estimation procedure. The spatial resolution of the KinectV1 depth data at 8 metres is still $<$20 cm [AJ+12]. The other limitation is the image resolution of the RGB camera. The maximum distance at which an eye-pair can be detected depends on the amount of pixels occupied by the eye-pair of the tracked person in the image. For a person with 60 mm pupil distance, using a 5-megapixel (2592$\times$1944 pixels) image taken with a camera with a 62° horizontal field of view, the maximum estimated distance, at which the person can be detected, is approximately 684 cm.

Thirdly, the fusion system is able to provide a binary measure of orientation, which means the system can tell whether a user is looking towards the sensors or not. It is not possible to obtain this from the KinectV1 skeleton data as it only provides a single point for the head joint. The presented approach makes it possible to design interfaces, which can, at the least, filter out likely passers by based on their head orientation. *SpiderEyes*, presented in Chapter 6 supports exactly this functionality. The system can distinguish between people actively viewing the system and people that are casually passing by or are standing in the background, engaged in other activities. This makes the system more practical in open office and large laboratory environments. In general, systems that are able to separate "attentive signals" from background signals have a distinct advantage in terms of being able to distinguish between active users and non-users in real-world deployment.

Since this version of the tracking system was developed, a new version of the Kinect sensor was released. It will be referred to as KinectV2 in this discussion. The second version of the Kinect sensor improves on the majority of the performance metrics of the original Kinect. It supports skeleton tracking of up to 6 people simultaneously at distances between 0.5 m and 4.5 m. Additionally, the skeleton tracking now includes joint rotation information, which means that head and body orientation can be tracked as well. The sensor also features a 1920$\times$1080 pixel RGB camera and the pixel resolution of the depth camera is higher too. This means that when comparing the CV-KinectV1 tracking system presented in this section to the KinectV2, most of the advantages of the system no longer apply. The only remaining advantage is the somewhat longer distance range at approximately 5 metres of confirmed maximum reach. However, the approach remains valid and the ability to use the computer vision only version of the algorithms with common RGB cameras when a depth camera is not available is valuable.

4.5 Conclusions and Recommendations

Sensors and tracking technologies are always improving. However, they are not always universally available or have a sufficiently low cost to be universally accessible. This chapter covers a technological platform that enables real-time distance sensing using commodity hardware and open-source frameworks.

The chapter starts with an overview and comparison of existing room-level tracking technologies. This is followed by details of design, implementation and evaluation details of a pure computer vision approach for distance estimation as well as coarse grained orientation detection. The limitations of this approach as well as lessons learnt from this approach are presented next. Additionally, a number of mitigation strategies and design improvements to address complexities with computer vision use in heterogeneous conditions is included. The next part of the chapter focuses on an extension of the computer vision approach to enable multi-person tracking and increasing distance estimation accuracy by coupling the computer vision approach with a depth camera.

The findings from this chapter can be distilled into a set of recommendations for designers of future interactive systems using low-cost tracking technology. Developing reliable tracking systems is a challenging task, especially when there is a constraint on low-cost high-availability hardware. Where possible, system designers should focus on leveraging already generally available sensor systems such as the Kinect (especially in version 2), which have a relatively low barrier to entry and still provide sufficient accuracy and general performance to enable prototyping of research systems. However, where the interactions to be examined fall outside the ideal sensor range, a more custom solution may be required.

If a sensor such as the Kinect is unsuitable, the algorithms introduced in this chapter may be recommended for either very short range interactions (say around a work desk), or at very long ranges, outside skeleton tracking range of the Kinect. The particular strength of the algorithms is that they leverage hardware resources already likely present in the interactive hardware (e.g. built-in web cameras). However, if very high (sub centimetre) distance sensing accuracy or more fine-grained orientation detection is required, the use of specialised sensors is recommended even after accounting for their higher deployment and maintenance costs.

To conclude, the algorithms and tracking systems presented in this chapter enabled successful deployment of the research prototypes presented in Chapters 5, 6, and 7. The technological platform has proven sufficiently robust to perform any needed evaluations, providing means to generate valuable research contributions.

Chapter 5

Case Study 1 - Interaction with a Multi-View Display

Chapter 3 demonstrated the variety of uses of spatial interactions in existing systems, revealing a number of possible opportunities for exploration. Chapter 4 presented the tracking algorithms that enabled distance, position and to a limited extent orientation sensing with common, high-availability hardware. The next three chapters, starting with this one, utilise the tracking platform in different ways to explore different physical, visual, and interactional properties of displays and how they can be leveraged to create novel interactions.

The focus of the case studies presented in this, and the two following chapters, was chosen to sample from as broad a spectrum of display properties as possible within the scope of this thesis. Therefore, each of the case studies concentrates on a very distinct goal. *MultiView*, the case study presented in this chapter, demonstrates that it is possible to exploit an otherwise limiting set of physical and visual properties of a specific type of LCD displays to create spatial interactions.

In Chapter 6, the *SpiderEyes* case study assumes a more traditional tracked space with a large shared display, potentially used by multiple people simultaneously. Given that the amount of display space is always limited, even with large displays, and especially so in a shared space with multiple users, the *SpiderEyes* case study explores the possibilities for dynamic manipulation of on-display visual content using a software approach, rather than a hardware one. The *MultiView* and *SpiderEyes* case studies concentrate on interactions with displays, where the display is the primary target of the visual focus of the people interacting.

The third case study, *DiffDisplays* (presented in Chapter 7), explores interactions taking place while the person interacting with the system is not looking at the display. *DiffDisplays* explores techniques for tracking visual change on unattended displays. To summarise, the first two case studies focus on examining different aspects of hardware and software approaches to creating novel spatial interactions with focus on the display, while the *DiffDisplays* case study uses spatial interactions to study occasions when the focus is away from the display.

The focus of the case study presented in this chapter is exploiting physical properties of displays. The most significant feature of the prototype systems is that they enable spatial interactions that do not require explicit use of tracking systems. This is achieved by taking advantage of the contrast and colour shifts typical for Twisted-nematic LCD displays (TN LCD). What can be considered a limitation of a particular display technology provides an opportunity for richer interactions by generating two simultaneous views of content on the display. In this chapter, a *view* is a set of visual content shown on a display (generally occupying the entire display), visible from a set range of angles. Figure 5.1 illustrates this visually, with each of the persons seeing a different view. Most displays only offer a single view from every angle. The systems in this chapter use a technique that

allows for two distinct, overlapping views of content to be shown on a single display. Each of the views is visible from a different viewing angle to the display.

While the main focus of this chapter is on exploring spatial interactions, which do not always require a tracking system, the two prototype systems also demonstrate how the inherent characteristics of interactive objects, such as the LCD displays, can be utilised simultaneously with a more traditional tracking approach. Both prototype systems also contain techniques which leverage information from a distance tracking system, further expanding the interactive possibilities.

5.1 Multi-View Displays

With the aim to couple viewing angles with spatial interactions, technologies enabling multiple views on the same display were examined. Most of these technologies use either spatial or temporal multiplexing. Systems using actively synchronised shutter glasses or passive polarised glasses employ displays (or projectors) with a high refresh rate to display interlaced frames for each eye. While the interlacing of the content is generally not noticeable with a sufficiently high refresh rate, the systems are very resource intensive and require user augmentation with glasses. Spatial multiplexing displays, such as those using lenticular sheets or parallax barriers do not require user augmentation but instead reduce the effective resolution of the displays. Dodgson [Dod05] provides a more detailed overview of the available multi-view technologies.

There is an alternative method for displaying multiple distinct views of a single display. Unlike the previous methods, it can take advantage of either temporal or spatial multiplexing and does not require user augmentation. It is based on exploitation of specific properties of TN LCD displays, first described by Harrison and Hudson in 2011 [HH11] and explored further by Kim et al. [KC+12] in 2012. The method relies on the specific compression of visible colour space of TN displays at different viewing angles to produce two distinct views using colour manipulation. With the availability of inexpensive TN LCD displays, this method lends itself for use in proof-of-concept prototypes such as the ones in this chapter.

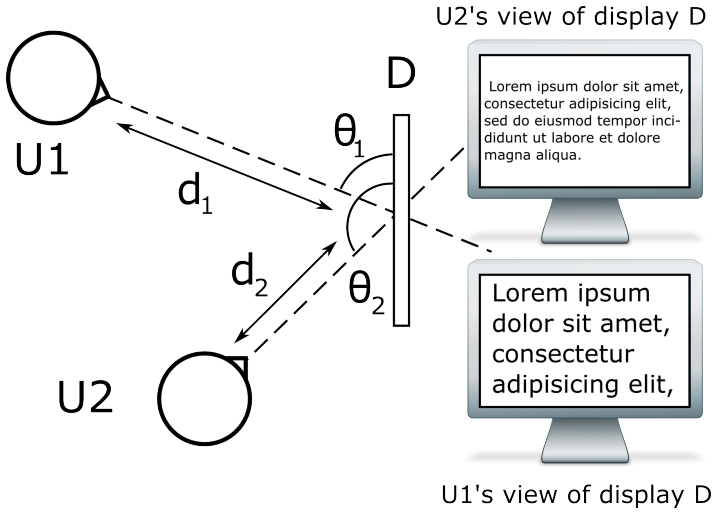


Figure 5.1: Two users viewing the same display with different distance-sensitive views from different angles. Each user sees different content — a different view.

5.2 Interacting with Multi-View Displays

The design space for applications using viewing angles, as well as distance, is potentially very rich. Three samples from the design space, each representing a different perspective, are provided below along with examples of applications and interaction techniques.

Using distance (e.g. d_1 or d_2 in Figure 5.1) between the interaction surface and a person provides opportunities for user interface alterations, or control along the axis formed by the line between the person and the surface. This is primarily explored in *SpiderEyes*, the case study in Chapter 6. However, adding knowledge of the angle of view to the surface (e.g. θ_1 or θ_2 in Figure 5.1) along the horizontal axis, and using a display capable of generating distinct views at different angles, allows for a much more fine-grained control. While this also applies to the vertical axis, making the person jump or squat to see a different view may not be practical. Visualisation of complex information is an example of where exploiting content modification using spatial properties along multiple axes could prove beneficial. Each of the different axes could be used to control a different aspect of the visualisation. For example, the distance could control the amount of detail visible, while the horizontal angle of view could be used to control movement through time within the dataset, and the vertical angle of view could provide alternative views of the data. This example would work equally well for a single person or multiple people.

Use in strictly multi-person scenarios is even more compelling. New sharing and collaboration techniques could be developed to take advantage of the additional information about location. With *Public Ambient Displays*, Vogel and Balakrishnan explored the notions of linking distance with expectation of privacy by exposing increasingly private information as people approached the display [VB04]. Imagine an extended version of the system that would dynamically shift between public, private or shared views for each user depending on their position and viewpoint overlap with other users. Different strategies for sharing such as distance-dependent screen splitting could be employed when the same view is shared by multiple people. When each person sees a different view, the views could be used for further, perhaps subtle, personalisation. For example, the size and shape of text and other visual content could be adapted based on the distance and viewpoint to optimise readability, which is similar to *Screenfinity* [SMB13]. However, using distinct views for each person could help avoiding some of the limitations present in the *Screenfinity* system such as content overlaps due to people's positioning or the need to compromise optimisations when multiple persons are close to each other but at different distances to the display.

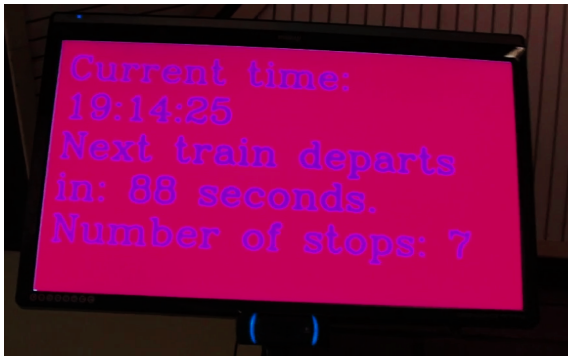
In order to demonstrate the possibilities for spatial interactions using multi-view displays, two practical applications have been chosen. The Video scenario uses multiple concurrent views to selectively display or hide subtitles (see Figure 5.3), whilst simultaneously adapting the size of the subtitles to keep their size constant from the viewpoint of the user. Subtitle hiding is a demonstration of a spatial interaction based on the viewing angle to the display (an Object-Actor orientation relationship) that does not require spatial tracking. The font-size manipulation that keeps their apparent size constant from the user's viewpoint uses active distance sensing to continually optimise the view of the subtitles (an Actor-Object distance).

In the Train scenario, the system simultaneously renders two separate views to people approaching the display from different distances (see Figure 5.2). With the first view (Figure 5.2a), the display shows a static set of information when viewed from a long distance away. The second view (Figure 5.2b) shows one of two distance-dependent interaction zones. The static view is visible independently of the two closer zones, again not requiring any active tracking. The two dynamic interaction zones use active distance tracking to alter the amount of detail visible. The benefit of the two distance-dependent views is that the apparent size of the text remains similar to the viewer, but the amount of displayed information visible in the closest view can be increased, as the apparent size of the display is larger due to the closer viewing distance. The Train scenario

exploits the Actor-Object distance relationship both actively (tracking-dependent distance-based switching between two of the interaction zones closer to the displays) and passively (with the static view always available to people viewing the display form a distance beyond the boundary of the actively tracked space).

The chosen application examples cover both single-user and multi-user cases and explore the usage of the angle of view on the horizontal (Video) or vertical (Train) axes. In order to examine the viability of using multi-view displays for spatial interactions using the two proposed scenarios, a formative validation study was conducted for each implemented prototype.

(a) User 1: Passive view visible when the user is far away from the display



(b) User 2: Active view visible at a medium distance from the display

Figure 5.2: The multi-view display observed by users 1 and 2 in the Train scenario. Images are for illustration only due to difficulty in capturing colours and contrast in real use on a digital camera.

(a) User 1: Passive view

(b) User 2: Active view with subtitles visible

Figure 5.3: The multi-view display observed by users 1 and 2 in the Video scenario. Images are for illustration only due to difficulty in capturing colours and contrast in real use on a digital camera.

5.3 Study Parameters

The two example application scenarios introduced above demonstrate possible applications for systems that alter the user interface based on the distance from the display as well as the angular viewpoint. An additional goal of this case study was to elicit people's opinion about the potential

usefulness of such systems in real-world scenarios. Since these are proof-of-concept systems, one of the design goals was easy replicability using consumer-grade technologies.

5.3.1 Multi-View Display

In order to generate two distinct views on a single display, a method presented by Kim et al. [KC+12] was adapted. Using a TN LCD display significantly reduced the cost of deployment compared with other technologies. A 24″ Iiyama ProLite E2407HDS display was used in the evaluation of the TRAIN scenario, while a 24″ Dell TN LCD display was used for the VIDEO scenario. The colour compression used to create the two views is most visible along the vertical axis of the TN LCD display. The two generated views were spatially multiplexed onto the display, essentially interlacing them. The two simultaneous views are best visible when viewed from relatively acute angles to the display. The views overlap when looking at the display near the centre of the viewing area. Figure 5.4 and the supplementary video for this chapter illustrate this visually.

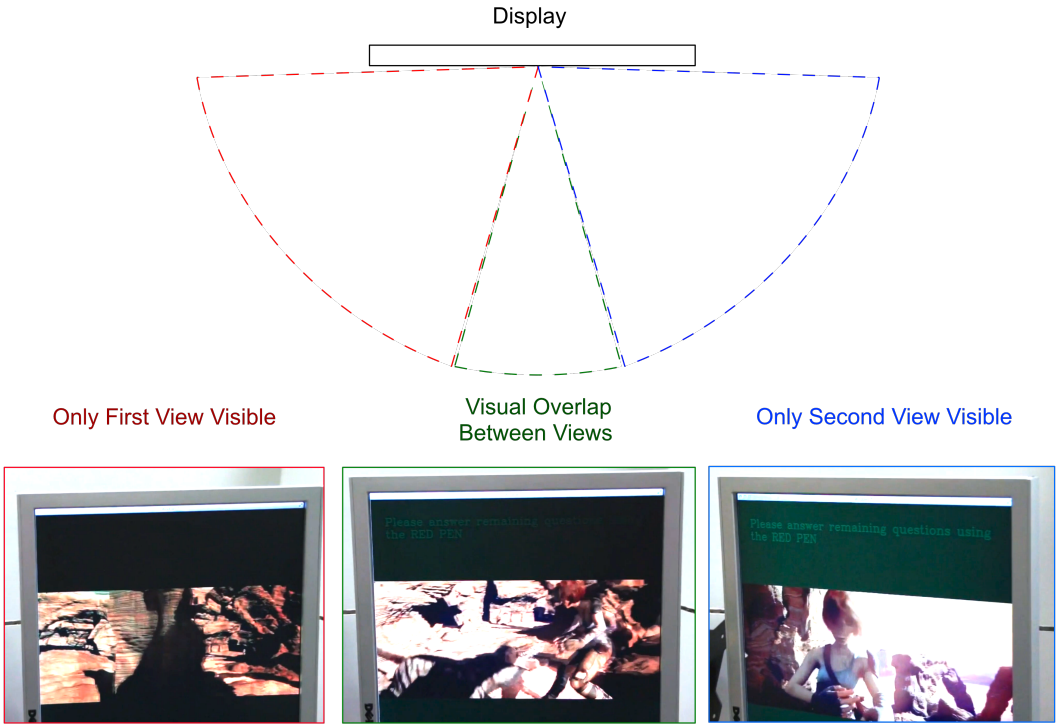


Figure 5.4: An illustration of the visual overlap of the distinct views when generated using the colour compression technique. The diagram uses a bird's eye view and includes examples of what the display looks like from each of the three angular zones.

The prototype in the TRAIN scenario has three interaction zones (as shown in Figure 5.6). However, as the two closest are distance-sensitive, the system only needs to show two distinct textual views (one for the two dynamic zones and one for the static zone). The display was in its default landscape orientation as the distinct views were intended to be located along the vertical axis. Primary colours were reported to provide the most contrast by Kim et al. [KC+12], so the focus was on those when choosing the colours of the text and background. Unfortunately, the multiplexing of the two views led to colour interactions, which changed the perceived colour and contrast when the two distinct views were shown simultaneously. This meant that while the text was readable as expected, the

overall colour scheme was not very pleasing to the eye. However, since the system achieved its design goals of generating two distinct views, the validation study continued with the colour scheme.

For the Video scenario, colour compression was used to hide subtitles, while affecting the displayed movie as little as possible. Unlike in the Train scenario, text only needed to be visible in one of the two views. This meant that it was not necessary to multiplex the views and thus the full resolution of the display could be used. Additionally, it was possible to choose the optimal colour in terms of contrast and subtlety. Using a specific shade of green (RGB(0,15,0)) as the background of the letterbox surrounding the movie and a brighter shade for the subtitles (RGB(0,90,0)) enabled the display of two distinct views approximately 40° apart. The LCD display was set to portrait orientation to place the two views along the horizontal axis, so that two people sitting next to each other would see different views.

5.3.2 Sensing Distance

While the primary purpose of the explorations in this chapter is to determine the viability of utilising the inherent properties of displays to enable spatial interactions without the need for a tracking system, each of the two prototype systems utilised a distance sensing system for additional interaction techniques.

In order to sense the distance of the people interacting with the prototype systems, the improved single RGB camera version of the tracking system described in Section 4.3.4 in Chapter 4 was used. The computer-vision-only tracking achieves the highest accuracy after a per-user calibration. However, using the RGB and depth camera fusion approach described in Section 4.4 of Chapter 4 completely eliminates this requirement and can be used as a drop-in replacement.

5.3.3 Participants

The same set of participants were used in both studies. 18 participants (10 female) were recruited among university staff and students (ages 20 to 36, mean 22.5, three wore glasses). Eight of the participants had a technical background ($>$ 1 year of Computer Science or related studies), while ten were from a range of other disciplines. No participants had been exposed to any similar system in the past.

A within-subject design was used. There were two conditions in each of the two studies, four in total. In order to counter balance bias due to possible learning effects, the ordering of scenarios and conditions of each user study was determined with a latin square. Before participation, participants were tested to confirm that each system was reliably capable of detecting their eyes and measuring distance. All tests were successful.

5.4 Scenario 1: MultiView Train Board

The idea behind the Train scenario was to create a system that introduces interactivity to a train station board, showing information about departing trains. Most current train station displays cycle through static information about the time of departure, departure platform and so forth. While this information is tailored to the display, it represents only a single view, which becomes increasingly sub-optimal as a person's distance to the display changes. At close distances, the view of the on-display information is skewed and the text can be too large for comfortable viewing. When the viewer is far away, the text is comfortably sized, but because the display has a relatively small apparent size at longer distance, the amount of information that can be displayed is severely limited. The prototype system presented here retains the existing comfortable viewing experience of a limited amount of information at larger distance. However, it addresses the issues of text readability and

available information granularity at closer distances by introducing two additional interactive views that display more detailed train information, while keeping the apparent text size comfortable even as the distance to the display decreases.

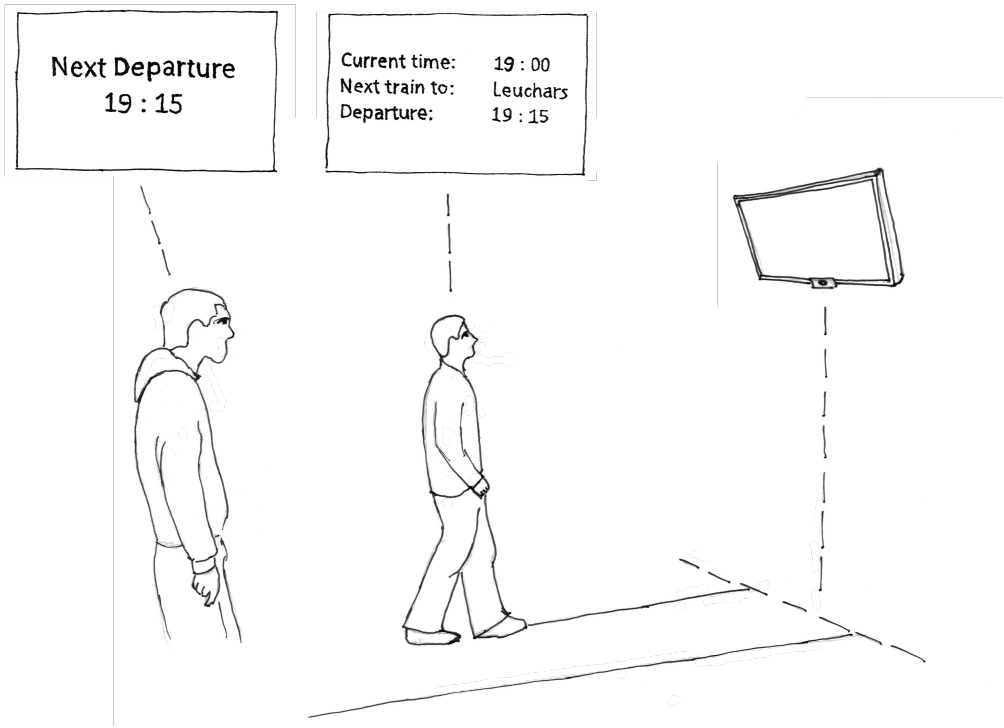


Figure 5.5: An illustration of the Train scenario. Each person sees different information simultaneously.

5.4.1 Procedure

The setup for the Train scenario is shown in Figure 5.6. The interactive space in front of the display is divided into three zones (Close, Medium, Static) using distance sensing and different angles of view. The Static zone was not interactive and it was created to replicate information on current displays at train stations. Visible from the static zone was a screen cycling through information about the current time, time of the next departure and the terminus of the next train. This zone was created by allocating one of the two viewing angles of the display and it was designed to be visible at distances greater than 250 cm. The Medium and Close zones were created using a combination of the second viewing angle of the display and our distance-sensing sub-system. The Medium zone was visible when the participant was between 125 cm and 250 cm away from the display and showed information about the current time, a timer to next departure and the number of stops the train would make. The Close zone was visible when the participant was between 0 cm and 125 cm away from the display and displayed a timer to next departure and a list of all the stops the train would make.

For each scenario, two conditions were tested—here referred to as the Active and Passive conditions. In the Passive condition no explicit user movement was required to complete a task, whereas Active meant a participant had to move to complete a task. For each condition, the

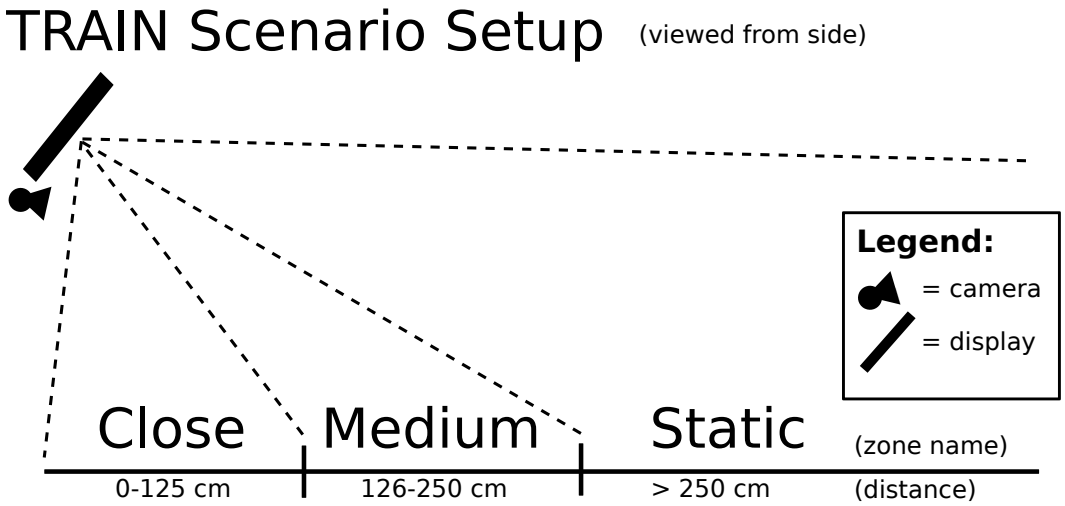


Figure 5.6: A diagram of the setup for the Train scenario. The Static zone does not use any distance sensing, while the display of the Medium and Close zones depends on the distance of the person closest to the display.

procedure followed a similar pattern. First, the participant was introduced to the task they would perform. In the Train scenario, the primary task was to gather information on departing trains.

The participants were also presented with a numbered sheet for answering questions relating to primary performance measures (questions related to information graphically or textually presented on screen). After completing their task, participants were asked to complete a questionnaire related to the secondary measures, verifying the functional performance of the system (e.g. readability of the text on a display). After both the Active and Passive conditions were tested, the participants were asked to complete one final questionnaire regarding the perceived qualities and usefulness of the application within these scenarios.

In each condition (Active, Passive), there were three trains leaving the station within the five minutes allocated to the task. In the Passive condition, each participant was asked to remain stationary at a point 5 metres from the display to ensure they saw the Static view. In the Active condition, participants were encouraged to move and explore the interactive space. In this condition, the questionnaire was designed to make it impossible to answer all the questions without moving between zones.

5.4.2 Results

To ensure users could accurately perceive the displays the correctness of the answers to the factual questions was assessed. An example of a question participants were presented with is *"How many stops does this train make?"*. Each participant was asked six factual questions for the Passive condition and twelve questions for the Active condition (18 answers per participant in total). The six additional questions for the Active condition were included to ensure the participants would have to use all three interaction zones. All of the resulting 216 answers were correct. This confirms that despite the contrast of the text not always being high, the text was always sufficiently readable for participants both in terms of size and contrast. This shows that the prototype system reliably demonstrated the required functionality, such as switching views between different interaction zones and adapting the text size.

After each tested condition, participants were asked to explicitly confirm the behaviour of the systems from their perspective (for example *"How many interaction zones did you notice?"*). All participants successfully observed all three different interaction zones (including changes in colour of text) in the ACTIVE condition. In the PASSIVE condition all participants except one only saw a single interaction zone. However, even this participant only noticed a faint shadow of one of the other interaction zones and was unable to make out any of the text. This confirms the multi-view system was usable in all three interaction zones.

Finally, for the question *"Do you consider this type of a dynamic dual view system useful?"* (1 = Very useful, 7 = Completely useless) the median rating was 3. For the question *"Would you use this system if it were installed at a real train station?"* (1 = Definitely Yes, 7 = Definitely No) the median rating was 2.5.

5.5 Scenario 2: MultiView Video Player

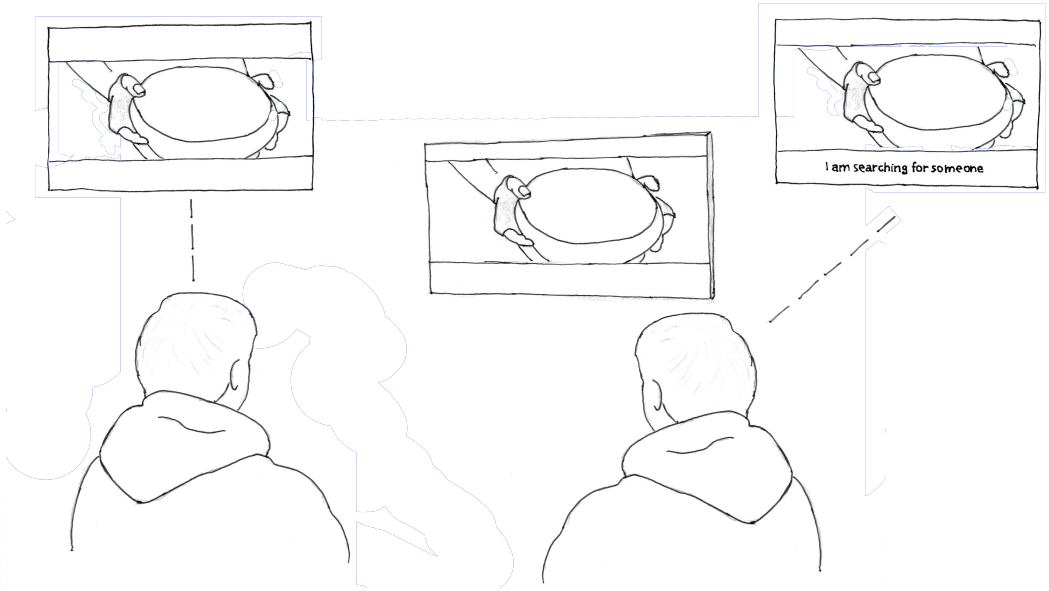


Figure 5.7: An illustration of the VIDEO scenario. The person on the right can see subtitles, while the person on the left cannot.

The use of subtitles can be crucial as they enable people that would otherwise have trouble following the movie to fully enjoy the experience. However, the presence of subtitles can have adverse effects on the rest of the group by potentially causing a shift in attention and creating distractions. In foreign language teaching, studies such as that by Lafiti et al. [LMM11] suggest that subtitles improve listening comprehension. However, Taylor [Tay05] found evidence that the added cognitive processing strain of reading subtitles can have a negative influence on some groups of foreign language students. The VIDEO scenario uses a system, which has the ability to show a video clip including subtitles to one part of the audience, whilst only showing the video to everyone else. In addition, the system can also dynamically modify the apparent font size of the subtitles depending on the distance of the viewers. This results in a multi-user, multi-view distance-sensitive movie experience, which addresses concerns raised by Lafiti et al. by showing subtitles to interested viewers, while also simultaneously accounting for Taylor's findings by allowing the viewers the choice of whether they want to see the subtitles or now.

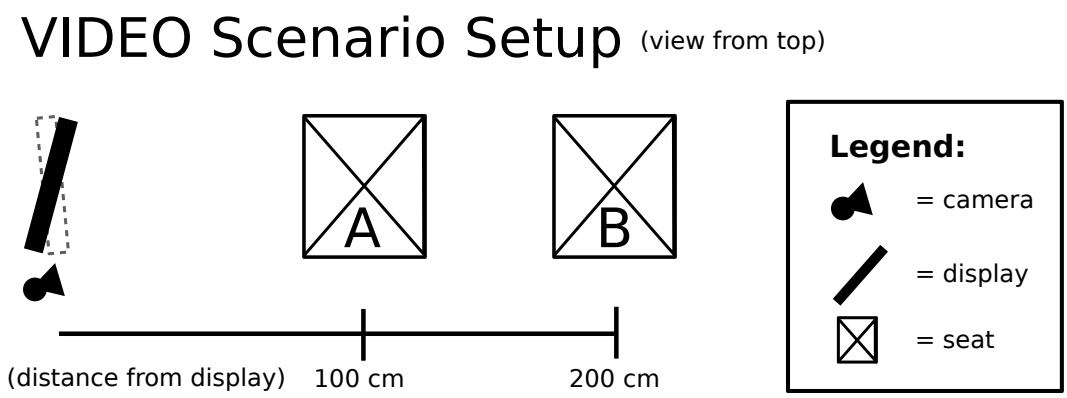


Figure 5.8: A diagram of the setup for the Video scenario. The display was tilted in different directions for each condition (dotted outline) to simulate a living room setup.

5.5.1 Procedure

The physical setup of the Video scenario is shown in Figure 5.8. Two seats arranged in two rows were used. In the Passive condition, participants were seated on seat B and the display was swivelled so that they would be looking at it at a 20° angle horizontally to the right from their point of view (see Figure 5.8). This angle was used to hide any subtitles without affecting the colour accuracy of the video much (effectively the primary view of the multi-view display). In the Active condition, the participants were initially seated on seat A and the display was swivelled 20° horizontally to the left from their point of view (see Fig. 5.8 for reference).

The Video scenario used the same protocol as the Train scenario, consisting of an introduction to the primary task, performance of the primary task and filling out an answer sheet to gather performance related measures. In both conditions (Active, Passive), the participants were instructed to answer a number of questions about the content in the video (six questions in each of the two conditions). These questions were not provided to participants right away but appeared on the screen while the video was playing. Questions related to what was happening in the video at the times they were shown. The participants were also instructed to answer questions in a black colour pen. The reason for these differences was that apart from the questions (as in Passive), they were expected to see subtitles under the video as well as a set of instructions complementing the questions shown above the video. Instructions included changing the seat to one closer or farther from the display, switching the colour of the pen when answering questions, and raising their hand to alert the experimenter.

After completing the primary task for one of the conditions, participants were asked to fill out a questionnaire verifying the secondary measures (e.g. visibility of subtitles). After completing the tasks for both the Active and Passive conditions they were administered a final questionnaire soliciting their opinions about the system.

5.5.2 Results

Similarly to the previous scenario, the answers to the factual questions the participants were presented with (e.g. *"What is the name of the dragon?"*) were checked for correctness. Participants were asked 6 questions per condition. Additional data points were gathered about whether or not the participants followed the on-screen instructions in the Active condition. Every question was answered correctly by every participant with two exceptions. One participant put one of their answers in the wrong place and answered another question wrong. Another participant consistently mistook their left side

for their right (three questions altogether). As with the TRAIN scenario this confirms that users were always able to accurately read the text.

The answers to the secondary measures confirmed the functional behaviour of the systems from the participants' perspective (for example *"Did you notice any subtitles?"*). Unexpectedly, one out of the 18 participants was able to notice the subtitles in the PASSIVE condition, but even then only faintly. In the ACTIVE condition, all the participants were able to see and read the subtitles and they successfully reacted to all text-based instructions. This confirms the expected behaviour of the system.

Finally, for the question *"Do you consider this type of a dynamic dual view system useful?"* (1 = Very useful, 7 = Completely useless) the median rating was 2. For the question *"Would you use this system at home?"* (1 = Definitely Yes, 7 = Definitely No) the median rating was 3.

5.6 Discussion

The two user studies verified that it is possible to design multi-view proxemic systems using commodity hardware and software. As a first exploration of the first multi-view spatially-aware systems implemented, the studies confirm that the systems provide sufficiently accurate information for users to solve a set of routine tasks using them. Users also report that they find such systems useful and would use them if they were available. After testing the two systems participants were invited to provide comments about the systems and the technologies. Comments were generally enthusiastic. Some representative comments are included below:

VIDEO scenario: *"Clever! I like the way it works towards an inclusive solution, not an exclusive one."* (Participant P18), *"I think it's a really good idea, especially for cinemas. I found the screen without the subtitles for me better because I found the subtitles distracting."* (P17), and *"It would be helpful if one person was hard of hearing as they could see the subtitles without distracting other people watching."* (P01).

TRAIN scenario: *"It is a good idea because if you are running into a station you can see basic info from a distance & as you get closer to the signs/platforms you then get the detailed info that you need."* (P11).

However, while the application prototypes successfully fulfilled all their requirements, designers should carefully consider the technological requirements of rich spatially-aware applications using viewing angles and multi-view displays. Designers are advised to use a different multi-view technology if multiplexing of different views is required. The results show that the colour compression method [HH11; KC+12] worked very well for the VIDEO scenario, which had no need to multiplex different views to generate the distinct simultaneous views. Using this method to create subtle interfaces that only aim to restrict the visibility of some of the information present shows potential. However, in the TRAIN scenario, the interaction between the colours led to textual views that were not very aesthetically pleasing. For scenarios such as these, the recommendation is to use alternative technologies not based on colour manipulation, such as lenticular sheets. Additionally, the colour manipulation-based method [HH11; KC+12] only allow separate simultaneous views along a single axis. Using lenticular lenses potentially allows generating simultaneous distinct views along both the horizontal and vertical axes.

5.7 MultiView and the IRE Model

This section provides a concise analysis of the two prototype systems presented in this chapter within the context of the IRE model introduced in Chapter 2. It mirrors the analysis that is already part of Chapter 3 in order to provide a clear, self-contained example of the kind of information that may need to be provided for successful analysis of future research systems. Corresponding sections are also present in the other two case study chapters.

	Distance					
Name	AA	AO	AE	OO	OE	EE
MultiView Train Board		D				
MultiView Video Player		C				

Table 5.1: Relationship - Distance

Relationship legend: A - Actor, O - Object, E - Environment; e.g. AO — Actor-Object
Value legend: C - Continuous adaptation; D - Discrete, zone or threshold based adaptation.

	Orientation								
Name	AA	AO	AE	OO	OA	OE	EE	EA	EO
MultiView Train Board					D				
MultiView Video Player					D				

Table 5.2: Relationships - Orientation

Relationship legend: A - Actor, O - Object, E - Environment; e.g. AO — Actor-Object
Value legend: C - Continuous adaptation; D - Discrete, zone or threshold based adaptation.

	Range			
	intimate	personal	social	public
Name	(<0.6m)	(<1.5m)	(<5m)	(>5m)
MultiView Train Board	x	x	x	x
MultiView Video Player		x	x	

Table 5.3: Interaction - Range

Value legend: x - System was described or demonstrated in use within this distance range.

	Cardinality					
Name	Actor				Object	Environment
MultiView Train Board	1	1*	M[1]	M*	1	
MultiView Video Player	1		M[1]		1	

[1] Only one person in the group can be actively tracked

Table 5.4: Interaction - Cardinality

	Mode								
Name	Actor				Object				Environment
MultiView Train Board	Sp	Vis	Sym		Sp	Vis	Sym		
MultiView Video Player	Sp	Vis	Sym	Ac	Sp	Vis	Sym	Ac	

Table 5.5: Interaction - Mode

Value legend: Sp - Spatial, Vis - Visual, Int - Intent, Sym - Symbolic, Ac - Acoustic

Let us start by briefly re-introducing the two prototype systems from this chapter. *MultiView Train Board* is a prototype system that exploits inherent properties of a specific type of an LCD display to show a distance and orientation dependent multiple simultaneous views of information. The display has three distinct views that can be shown to a person based on their distance to the display. The

	Intentionality				
Name	Actor	Object			Environment
MultiView Train Board	E+	E		A	
MultiView Video Player	E	E		A	

Table 5.6: Interaction - Intentionality

Value legend: E - Explicit, E+ - Explicit (with possible misclassifications), I - Implicit, I+ - Implicit (with possible misclassifications), A - Ambient

	Intensity				
Name	Actor	Object			Environment
MultiView Train Board	N	U		N	
MultiView Video Player	N	U	S	N	

Table 5.7: Interaction - Intensity

Value legend: U - Unnoticeable, S - Subtle, N - Neutral, I - Intrusive, D - Disruptive

view that can be seen from afar can be seen simultaneously with one of the closer views. This is because the prototype exploits a viewing angle specific behaviour of TN LCD displays, which allows to selectively show or hide content from certain angles. The switching between the two views closer to the display is based on the distance of the closest person to the display, which is established with a computer-vision based tracking system.

MultiView Video Player is another system that uses a combination of vision based tracking and inherent properties of TN LCD displays to show distinct simultaneous views. In this case, however, the simultaneous views are used to selectively display additional content during video playback. People sitting on one side of the display see a video with subtitles, whereas people siting on the other side of the display only see the video and no subtitles. The size of the font for the subtitles is based on the distance of the viewers on the side of the display where subtitles are visible.

5.7.1 Entities and Relationships

In both of the prototype systems only Actor and Object entities are used. In both cases, Actors are the people interacting with the system and Objects are the displays being interacted with. Neither of the systems makes any use of an Environment as an interactive entity.

In terms of spatial relationships, both systems use distance and orientation between entities for interactive purposes. Neither of the systems uses the position relationship. *MultiView Train Board* uses distance to show different information or different amounts of information to the users of the system. Since the determining factor is the distance between a person (Actor) and the display (Object), it is classified as an Actor-Object distance relationship. Because there is a clear boundary between the two closer views, the relationship is classed as a zone-based adaptation. The vertical orientation of the display to the person interacting with it is used in the *MultiView Train Board* system to show only one of the two possible simultaneous views. This is a Object-Actor orientation relationship within the IRE model. Since there are only two views with a boundary between them, the relationship is a discrete, zone-based one rather than continuous.

MultiView Video Player also makes use of both distance and orientation and the classification is even more straightforward. The orientation of the display (Object) to a person (Actor) looking at it determines whether the person will be able to see the subtitles accompanying the video, this forms an Object-Actor orientation relationship. The relationship is classed as discrete as there are only

two zones, one with subtitles visible and one without. The distance relationship is an Actor-Object distance relationship and it is continuous as the size of the subtitles is continuously adapted to keep their apparent size constant based on the distance of the person (Actor) closest to the display (Object) in the area where subtitles are visible.

For clarity, the values for the spatial relationships are also shown in Tables 5.1 and 5.2.

5.7.2 Interaction

Looking at the measures relating to interaction, most of their values can be easily established. The most straightforward is range (values shown in Table 5.3). As Figure 5.6 and the supplementary video for this chapter illustrate, interactions with *MultiView Train Board* are possible at distances between 0 cm from the display (directly under the display) to well beyond the 5m distance (the maximum distance is only limited by the ability of the person looking at the display to read the displayed text). Therefore, the system allows for interactions at all the possible distance ranges. For *MultiView Video Player*, Figure 5.8 shows that the system has been used at distances between one and two metres from the display. This means that the system uses the personal and social ranges.

The values for the cardinality measure are shown in Table 5.4. The analysis of Object entities is straightforward as there is only ever a single display for both *MultiView Train Board* and *MultiView Video Player*. Actor cardinalities are somewhat more complex. With both systems, group and parallel interactions are enabled mostly by the design of the system rather than the tracking system. For *MultiView Train Board*, a single Actor can interact with the system. Multiple independent Actors can also interact with the system easily due to the system being able to show multiple independent views at different distances. There can be a group of persons at each of the distances, so group interaction is possible. However there is a limitation that only one person in the group closest to the display will be actively tracked. *MultiView Video Player* also caters for single Actor and group use. However, the scenario of watching a film limits the possible cardinality values. A single Actor can watch the video or a group of actors. Independent parallel use is not possible because only one video track can be played at a time. Similarly to *MultiView Train Board*, only the person closes to the display within the area where subtitles are visible will be actively tracked to determine the subtitle size.

In terms of modes (values shown in Table 5.5), both systems use the Spatial, Visual and Symbolic modes for interaction. The use of the Spatial mode is demonstrated by the spatial interaction between the people and the display. The use of textual information in both systems demonstrates the Symbolic mode and the fact that this information is communicated visually through a display shows the use of the Visual mode. The *MultiView Video Player* system also makes use of the Acoustic mode, but that is mostly a side-effect of the scenario in which the system is used (showing videos that include an audio track) rather than an explicit design decision to support specific interactions. Note that since the mode values capture communication between entities, the values are mirrored between the Actor and Object entities.

Table 5.6 shows the values for the intentionality measure. Once again, both systems are very similar. For Objects, the values are *explicit* and *ambient* for both systems. The display output is clearly *explicit* as the displays are always trying to explicitly communicate with the people interacting with them when active interactions are taking place. The *ambient* value is somewhat more subtle. An *ambient* action is essentially a completely passive state on the part of the interactive Object. In the case of *MultiView Video Player*, the separation of the views through orientation falls into this category. The display is not actively changing the view in any way, it is completely passive in deciding whether a particular person will see the subtitles or not. With *MultiView Train Board*, this *ambient* state is demonstrated by the static view from afar. Again, the display is completely passive in showing the view. Actor actions are explicit with both systems. The changes in distance or orientation towards the display are explicit actions on the part of the person interacting. The difference is that *MultiView*

Train Board also makes the assumption that a person approaching close to the display is expressly requesting more information when deciding whether to switch between the two active views closer to the display. This assumption can lead to misclassification of the intentionality of the action taken by the person (e.g. if the simply moved to get a better look at the display rather than to actively request a different view). Therefore, the Actor value for intentionality is classified as *explicit+* to account for this potential for misclassification.

Lastly, the intensity values are shown in table 5.7. For Actor entities for both of the systems, the intensity is classified as *neutral* because while the actions have an impact on the system driving the display, they do not interrupt the system in any way. For Object entities, the two systems differ slightly. The exclusive nature of the simultaneous views used by both systems means that some of the system's actions will be completely *unnoticeable* to some of the people interacting with the system, depending on their positions. Otherwise, most other actions by the displays are noticeable but non-intrusive, therefore classified as *neutral*. For *MultiView Video Player*, the changing of the size of the subtitles is classified as both *subtle* and *unnoticeable* because the participants of the evaluation only noticed the changes in size sometimes and often were not sure whether the size changed at all. This demonstrates that the importance of explicitly gathering information about the effect of actions directly from the interpreters of the actions.

To conclude this example analysis, I would like to highlight the need for careful consideration of the amount of information required to fully analyse a system using the IRE model, even including the design of evaluations. Taking the example of the subtitle size changes in *MultiView Video Player*, it would not have been possible to establish the intensity values for this aspect of the system.

5.8 Conclusions

This chapter contains the first of three case studies, exploring different aspects of interactions around displays using spatial relationships. This *MultiView* case study focuses on exploiting the physical and visual properties of displays, namely the limited viewing angles of TN LCD displays and the subsequent colour and contrast shift, to enable novel spatial interactions that do not require active spatial tracking. The two prototype systems presented in this chapter both utilise the same underlying technology, but they use it to leverage different spatial relationships.

The *MultiView Train Board* makes use of the multiple simultaneous views to enrich interactions based on the Actor-Object distance relationship by providing a static view, which is always visible to people at large distances from the display, while also including two additional distance-based views, which show more detailed information at comfortable text sizes to people closer to the display.

The *MultiView Video Player* utilises the two available simultaneous views for a somewhat more subtle, even unnoticeable, purpose (using the terminology for interactional intensity from the IRE model in Section 2.3.4 in Chapter 2). The prototype selectively displays and hides subtitles during video playback based on the orientation of the display to the viewer (Object-Actor orientation). The viewers in one part of the viewing space are not even aware that viewers in another part of the viewing space can see subtitles. Continuing with the subtle and possibly unnoticeable actions, the system also dynamically alters the text size of the subtitles to keep their apparent size stable from the viewpoint of the people able to see the subtitles.

The results of a formative evaluation indicate that the multi-view, multi-user spatially-aware interactive prototypes have been realised and operate as expected. The very accurate answers from the 18 participants indicate that both the distance and viewpoint sensitive aspects of the interaction support delivery of information. Furthermore, the interaction zones can be easily discovered and they were explored by all participants to complete the experimental tasks. Importantly, the expectation is that use of similar systems will primarily be in scenarios with multiple users, with distinct views being generated simultaneously for different viewing angles. In addition, spatial tracking is not

always required for all spatial interactions. The static zone in the Train scenario can operate without tracking, while the close and medium zones have the required tracking data to deliver dynamic, distance-sensitive information and updates. The same is true for the Video scenario, where tracking is not required for the orientation based subtitle display, while the subtitle size manipulation makes use of active tracking. Support for both types of spatial interactions is a positive feature of the presented approach. This is emphasised by the validation of the example systems and people's positive experiences.

Chapter 6

Case Study 2 - Prototyping Multi-User Interactions with a Large Display

This chapter collects and maps out the major content manipulation methods that have been coupled with spatial interactions to enable their use in single user or collaborative scenarios with large wall displays. This is followed by an exploration of how the different methods can be applied together within interaction scenarios, taking information visualisation as an example application. Finally, this chapter presents a simple prototyping tool, which leverages the tracking algorithms previously presented in Chapter 4 to enable rapid implementation of any combination of the collected content manipulation methods.

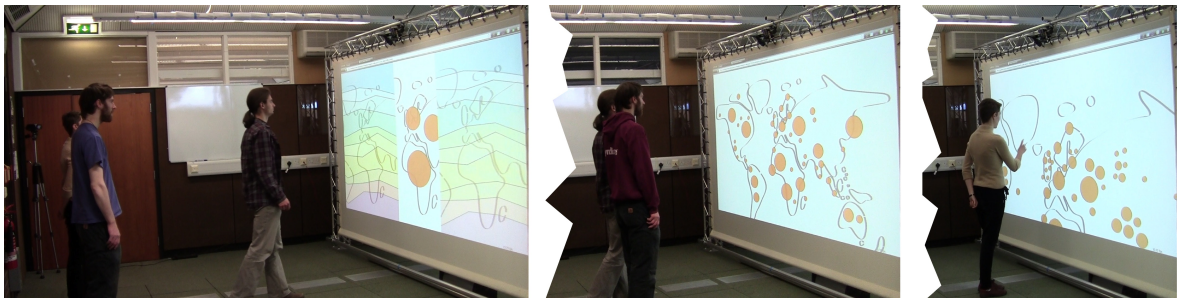


Figure 6.1: Three examples of single- and multi-user scenarios in SpiderEyes. The first shows three users working in parallel. The second shows group collaboration. The third shows exploration by a single user.

Chapter 3 showed the great variety of spatial relationships used by existing systems as well as some of the potential uses in future research explorations. However, the primary focus has been on the relationships rather than on their application. This case study is a limited, highly focussed systematic exploration into the design space of applications of spatial interactions, concentrating on visual manipulation of on-display content using spatial relationships.

In order to limit the scope of this chapter to manageable size, any results of the explorations will primarily be demonstrated using distance-based relationships, specifically distance of persons to a large display or from each other. Figure 6.2 shows this visually. The scenarios used to demonstrate the concepts presented in this Chapter are examined as multi-user scenarios as those are deemed to provide the best platform for the explorations. However, both collaborative and parallel interactions are supported and the scenarios are considered from a single-user perspective as well.

6.1 Introduction

The Chapter starts with an overview of existing literature. In contrast to Chapter 3, the literature review presented examines systems through the lens of manipulation of on-display visual content. The literature is leveraged to extract, which parameters of the on-display content are visually manipulated. Once the parameters are established, they are used to derive a matrix of possible combinations of content manipulation parameters. This matrix motivates a scenario based exploration of some of the previously unexplored combinations of content manipulation parameters. The figures for the scenarios were sketched by Uta Hinrichs with modifications by the author of this thesis. Uta also contributed to the conception and description of the interaction scenario.

The next part of the chapter concentrates on linking the observations made during the design, implementation and testing of the exploratory scenarios with existing research into collaborative interactions around large displays. This results in a set of design considerations for developers of future spatially-aware single- and multi-user applications, which visually manipulate content on large displays.

The last part of this chapter presents a prototyping tool, which enables rapid implementation of visualisation applications that utilise any combination of the content manipulation parameters using distances between people, and between people and a display. This is followed by an exploration of the use of the prototyping tool beyond distance relationships and content manipulation, instead using position in relation to the display to control an exploration of a visualisation of a multi-dimensional dataset.

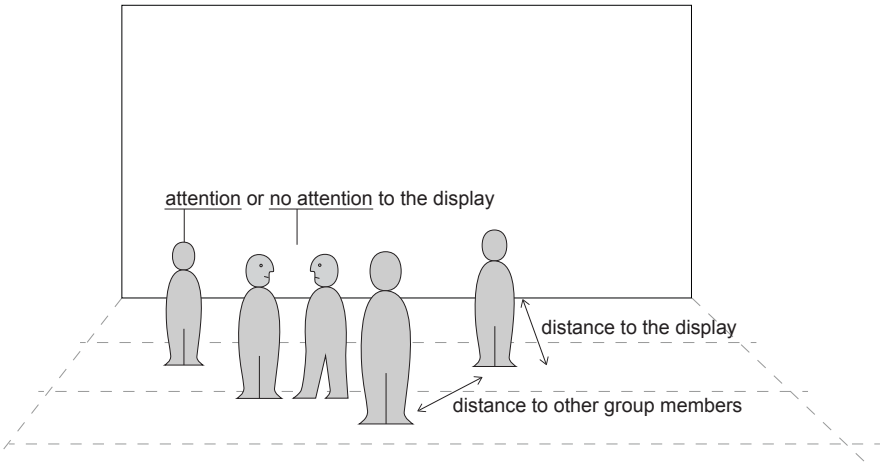


Figure 6.2: This Chapter primarily utilises distance from displays and between people to demonstrate findings. At times, this is augmented by some use of orientation as a proxy for attention towards or away from the large display. (Figure by Uta Hinrichs)

6.2 Content Manipulation on Large Displays

This section provides an overview of content manipulation techniques used in interactions with large displays in existing literature. The literature overview is then used to derive a matrix for systematic exploration of possible combinations of content manipulation techniques. This exploration includes classification of existing systems within the derived matrix. The gaps in the matrix are used to motivate a number of interaction scenarios exploring possibilities for novel interactions in both single- and multi-user settings, which are covered in the following section.

6.2.1 Content Manipulation on Large Display in Existing Literature

The following presents a summary of how knowledge about people's spatial relationships has been previously applied to support individual and social interactions around large displays. One of the most direct parameters that can be used to drive spatial interactions with a wall display is the distance of people to the display. Previous work has described a variety of techniques and example scenarios of how the distance of people to a stationary display can drive interactions. This previous work is classified here in terms of how information presented on the display is changed as a result of people's varying distance to the display.

Adjusting the Magnification Factor of Information Previous systems have adjusted the size or magnification factor of information as people move closer to or away from the display. *Lean & Zoom*, for instance, magnifies display content as people lean in closer to the display [HD08]. The actual display content is not modified but merely enlarged as a person leans forward. *E-conic*, as part of its perspective correction technique kept the apparent size of applications constant from the viewpoint of the tracked person [NS+07]. *Screenfinity* also included a technique to keep the apparent size of content constant [SMB13]. *Egocentric ZUI* amplified natural magnification caused by physical movement to increase the magnification available in a limited amount of space [RJ+13]. In *Proxemic InfoVis*, Jakobsen et al. [JS+13] explored two techniques amplifying magnification through physical movement, using either distance from the display, or relative distance from a designated area in front of the display.

Adjusting the Displayed Detail of Information An alternate way of using distance for changing the displayed content is to show different levels of detail depending on the distance to the display. This idea can be considered distance-based semantic zooming: more detailed information is added to the display as an individual moves closer. One variation of *Lean & Zoom* blends in additional labels to a 3D model as a person leans forward towards the display [HD08]. Jakobsen et al. [JS+13] explore this idea in more detail within the information visualisation context, where distance of a person to the display is used for aggregating shown data.

Adjusting the Amount of Information Presented Distance to the display has also been used as a parameter to adjust the amount of information displayed. For example, in *Proxemic Media Player* Ballendat et al. [BMG10] present a media player application where additional movie options are shown as an individual approaches the display. Arguably this is simply a function of magnification as the apparent size of the content area of the display increases as a person approaches closer to the display if the apparent size of the displayed content is kept constant. However, it is worth pointing out as intentional use of this effect could be made.

Adjusting the Type of Information The type of information can be adjusted based on people's distance to the display, an approach that some previous research has explored. Vogel and Balakrishnan [VB04] discuss a large display application for public environments (*Public Ambient Displays*) in which information becomes more personally relevant, and even private, the closer an individual is to the display. The idea is that more generally relevant public information is visible to people standing at a distance from the display, while people standing close to the display can bring up personal information, such as their calendar, and their body will shield this information from the view of onlookers.

Adjusting the Impact Area of Interactions In the context of multi-user interactive whiteboards, the technique "Field of View" developed by Seifried et al. [SR+12] takes people's distance to the

display into account to determine the horizontal impact area of undo/redo actions: being closer to the display results in a smaller estimated field of view and, therefore, a narrower impact area. The impact area affects content by limiting possible interactions. It does not explicitly affect the visual representation of the content, and therefore this dimension is out of scope of this exploration.

Drawing on work detailed in Chapter 3 and the analysis above, we can clearly see the varied use of spatial relationships to modify how content is displayed on large displays. In order to provide a better grounding of explorations into these techniques, a conceptual map of possible technique combinations is introduced.

6.2.2 Content Manipulation Matrix

Existing systems and interaction techniques provide a good starting point for mapping out the design space for content manipulation techniques, even though they do not represent an entirely complete set of possible techniques. In this section, the possible values for each dimension identified in existing literature are mapped out and clustered into categories. There are essentially four major aspects of content that can be manipulated - size, detail, representation, data, and complex manipulations, ordered approximately in order of likely visual impact. Table 6.1 provides a concise visual overview of the Content Manipulation Matrix, which maps out the content manipulation dimensions and mappings, as well as values for each of the systems from related literature.

Size

Depending on the distance to the display the magnification of content can be adjusted. Since altering the apparent size of content on the display is the most common content manipulation technique and there are a number of ways the size of content can be manipulated, all known variations are presented in some detail:

Physical Zoom. The content does not actively react to people's movements in front of the display but retains a constant physical width and height at all times. However, people's proximity to the display naturally increases or decreases the apparent size of information represented in the visualisation layer.

Constant Zoom. The viewing angle of the content, and thus its apparent size, is kept constant no matter how close people are to the display. That is, the size of the visualisation layer (width and height) is actively changed as people move back and forth in front of the display in such a way that the perceived size of the represented information remains constant at all times (see Figure 6.4 and Figure 6.8).

Amplified Zoom. The visualisation layer is scaled up or down depending on people's proximity to the display, amplifying the magnification impact of people's physical movement. As people move closer, the visualisation layer enlarges to a higher degree than with *Physical Zoom*, providing a more magnified view on the information represented (see Figure 6.5).

Inverse Zoom. This is a variant of *Amplified Zoom*. However, instead of amplifying the magnification impact of people's physical movements, the impact is dampened (or even reversed, depending on the dampening factor), decreasing the apparent size of the content as people move closer.

Detail

Manipulating the amount of detail shown covers both sampling and aggregation of data. The main reasons for varying the amount or granularity of the displayed content include dealing with clutter, perceptual limits (display resolution or visual acuity), and preventing information overload. Additionally, aggregating data using distance can provide valuable insights into the datasets as Jakobsen et al. [JS+13] demonstrated in their second study, which used distance based aggregation

		Content Manipulation				
		Size Only	Detail	Representation	Data	Complex
Size	Physical	-	LZ, PI, PA	*S1*	*S1*	*S5*, PM
	Constant	EC, SF, PM	*S4*	*S2*	*S2*	*S5*, PM
	Amplified	LZ, PI	*	*S3*	*S3*	*S5*, PM
	Inverse	EZ	*	*	*	*S5*, PM

Table 6.1: Content Manipulation Matrix is a grid map of the combinations of content and view manipulation methods, together with scenarios, which cover them in detail. Scenarios explored within this Chapter are marked S1–S5 and values for other systems gathered from related literature are also marked using an abbreviated form of the system's name (LZ = *Lean & Zoom*, EC = *E-conic*, SF = *Screenfinity*, EZ = *Egocentric ZUI*, PI = *Proxemic InfoVis*, PM = *Proxemic Media Player*, PA = *Public Ambient Displays*). Stars (*) mostly denote combinations that were not used by any systems in related literature or explicitly explored in one of the scenarios.

to show three levels of detail granularity: individual homes (dataset granularity), postal districts (aggregate), and municipalities (also aggregate). See Figure 6.8 for a visual example. A scenario primarily manipulating the granularity of the data or the level of detail shown is referred to as *Detail-Active*. While the distinction between public and private information is important, it is still an example of sampling of content, just with a specific sampling method. Therefore, the content manipulations in *Public Ambient Displays* fall into the detail category.

Representation

Depending on the distance to the display the visualisation type can be adjusted, providing a different view of the dataset. For example, people far away from the display can see the temperature layer. However, as they get closer to the display, the temperature layer can be replaced by a commodity cluster visualisation. Directly in front of the display people can see a commodity word cloud (see Figure 6.3 for visual illustration). A scenario primarily manipulating the representation of the data is called a *Vis-Active*.

Data

Rather than manipulating the level of detail or the visual representation of a dataset, the dataset itself can be changed. For example, consider a visualisation of stock price movements, where a person far from the display sees the visualisation of the price of company A, moving closer to the display changes the price dataset to that of company B, and moving closer yet will show the price for company C.

Complex

This category is a catch-all for all content manipulations, which are more complex than a combination of size manipulation and one of the other types of manipulations. This could either be a complex setup, which includes a *Detail-Active* layer at close distance, followed by a *Vis-Active* layer further away, or a set of layers, where each layer modifies datasets, representations and levels of detail simultaneously.

Discussion on Classification

The Content Manipulation Matrix 6.1 shows the values for the different categories of content manipulation for both the existing systems and the example application settings outlined in the

next section. In terms of notation, values *S1* to *S5*, in italics, denote scenarios explored later in this chapter. Other two letter values correspond to names of existing systems classified within the content manipulation matrix (the table caption includes a legend of full system names). The Physical/Size-Only cell has intentionally been left blank as this corresponds to standard configuration with no content manipulation taking place. Any system, which does not perform any content manipulation of the kind described in this chapter would be classified into the Physical/Size-Only category.

In terms of existing systems, it appears that most of them utilise only size content manipulations or detail granularity content manipulations. The only exception is *Proxemic Media Player*, which combines multiple content manipulation methods in multi-user scenarios. In single-user scenarios, only *constant zoom* seems to be applied. While it is likely that representation manipulations are somewhat uncommon outside of information visualisation scenarios, all combinations of content manipulations present opportunities for future research. Even though the scenarios presented in the next section do not cover all the possible combinations, they clearly demonstrate the potential for novel interactions.

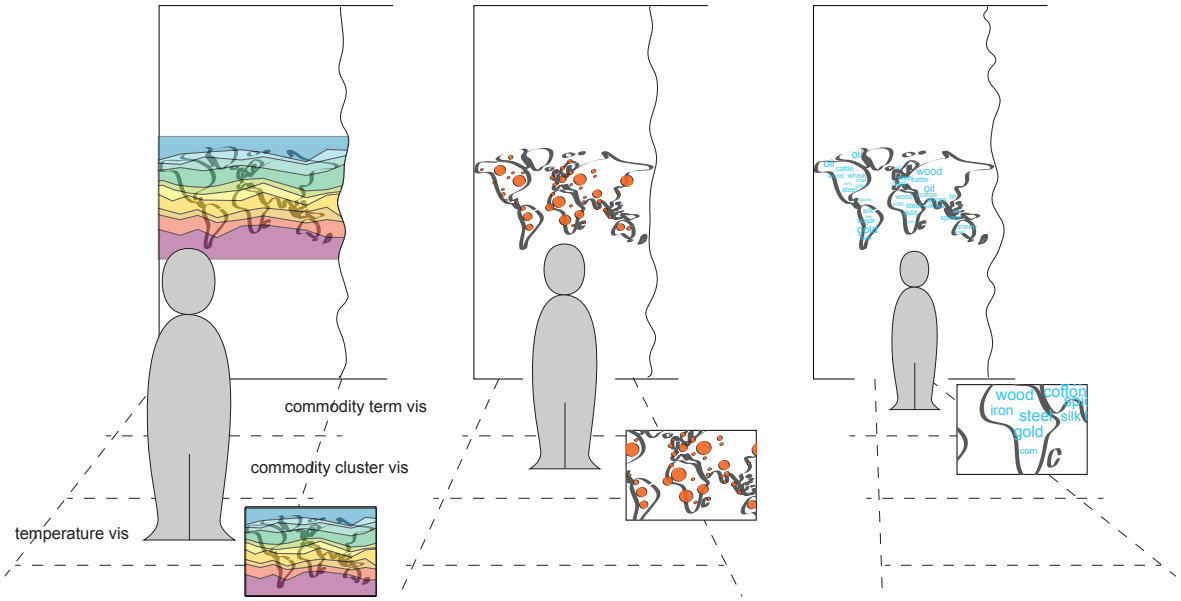


Figure 6.3: *Vis-Active* Display: proximity to the display determines the type of visualisation. Inset images towards the bottom of the figure show what the display looks like from the viewpoint of the person. (Figure by Uta Hinrichs)

6.3 Scenario Exploration

As the Content Manipulation Matrix demonstrates, content manipulation parameters can be combined in different ways, resulting in seventeen different possible combinations. Five of the scenarios, which have not been explored by any of the existing systems are described in this section in detail to point out the potential advantages and disadvantages of the parameter combinations, including their use in single- and multi-user settings. In order to demonstrate the content manipulation techniques and related concepts, a sample application scenario has been devised. This scenario will be used throughout the rest of the Chapter.

The interaction scenario is set in the context of knowledge work and visual analytics. Many data sets contain a number of facets or parameters that have to be analysed individually and in correlation in order to make sense of potential patterns and, eventually, to establish general insights. Consider, for instance, historians exploring the potential environmental impact of commodity trading in the 19th century. As part of their research they may analyse temperature changes in certain geographic regions and correlate these with trading activities of certain commodities in particular countries. This type of analysis may require different visualisation layers: a map that provides the geographic context for both the trading and the temperature data, a visual layer that shows the temperature in certain geographic regions (temperature bands), a visual layer that shows the frequency of trades in certain geographic regions (trading clusters), and a layer that shows the frequency of particular commodities in certain locations (commodity cloud). In search of potential patterns within the data, the historians have gathered around a high-resolution wall-sized display to explore the data together.

Depending on how the data architecture supporting this scenario is designed, the *Vis-Active* setting described below could be classified as either manipulating the representation of the content (if all the data was in a single complex dataset) or as manipulating the data (if each of the visual layers used a different dataset), or a combination of both.

6.3.1 Interaction Settings

Scenario 1: Vis-Active with Physical Zoom On the *Vis-Active* Display (see Figure 6.3), the visualisation layer changes based on an individual's distance to the display while the physical size of the visualisation remains constant. The mapping between proximity and visualisation type can be continuous with the visual layers blending into each other as people move towards and away from the display.

In a multi-user setting (see Figure 6.6) the visualisations change as group members move toward and away from the display. One of the possible advantages of this setup is that it supports independent explorations by users: each user can easily shift between visualisation views, and the continuous blending of visualisations even allows each user to explore correlations within the different data sets and perspectives. A possible disadvantage is that in collaborative situations, users cannot easily blend different types of visualisations while standing next to each other because they would need to be at different vertical distances from the display. According to findings by Hawkey et al. [HK+05], forcing group members to position themselves at different vertical positions in front of the display may hamper communication and coordination, which are important factors in more closely coupled collaborative work phases.

Scenario 2: Vis-Active with Constant Zoom In Scenario 2 the type of visualisation layer is adjusted as people move back and forth in front of the display while adjusting its size to keep people's viewing angle constant.

As shown in Figure 6.4, more context information can be added to the display as a person moves closer—because the viewing angle remains constant as a person moves toward the display, the (physical) size of information in focus does not change. This can facilitate direct interaction with information when close to the display because information is visible in a constrained space; people do not have to reach far or crouch to manipulate particular data of interest, something that has been reported as problematic on large wall-sized displays that show map representations [HK+05].

In a multi-user setting of the *Vis-Active* with *Constant Zoom* scenario (see Figure 6.7), users standing at different distances from the display will see different types of visualisation layers. At the same time, their viewing angle remains stable as they move towards or away from the display. Similarly to Scenario 1, in a collaborative scenario, individual users can work on different data perspectives and explore different visualisations at the same time, while each group member can

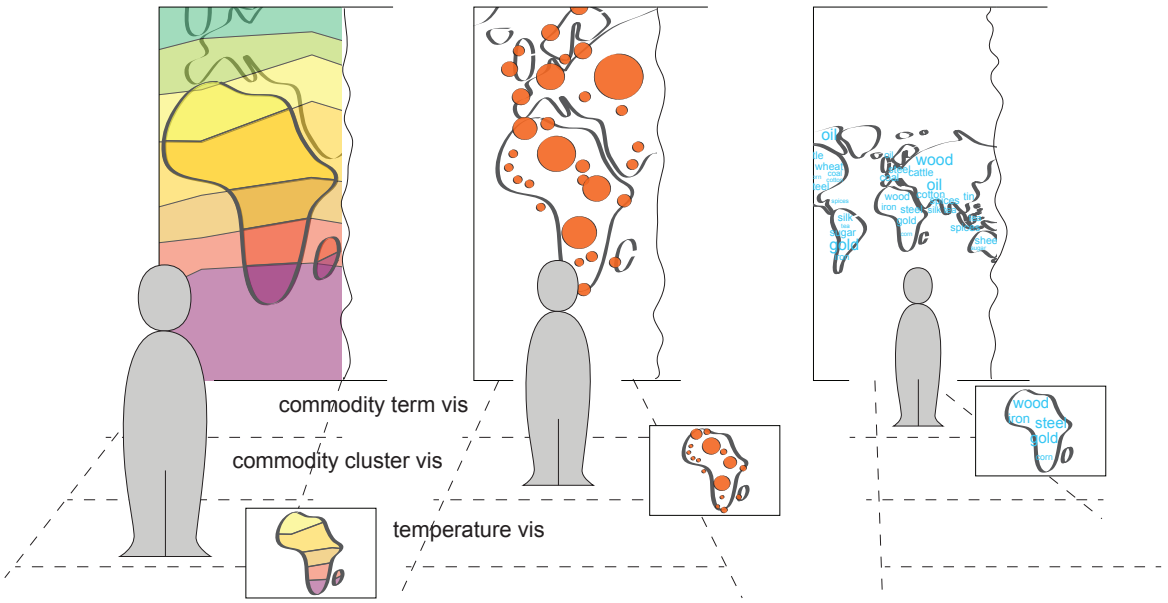


Figure 6.4: *Vis-Active* Display with *Constant Zoom*: the distance to the display determines the type of visualisation while people's viewing angle remains stable. (Figure by Uta Hinrichs)

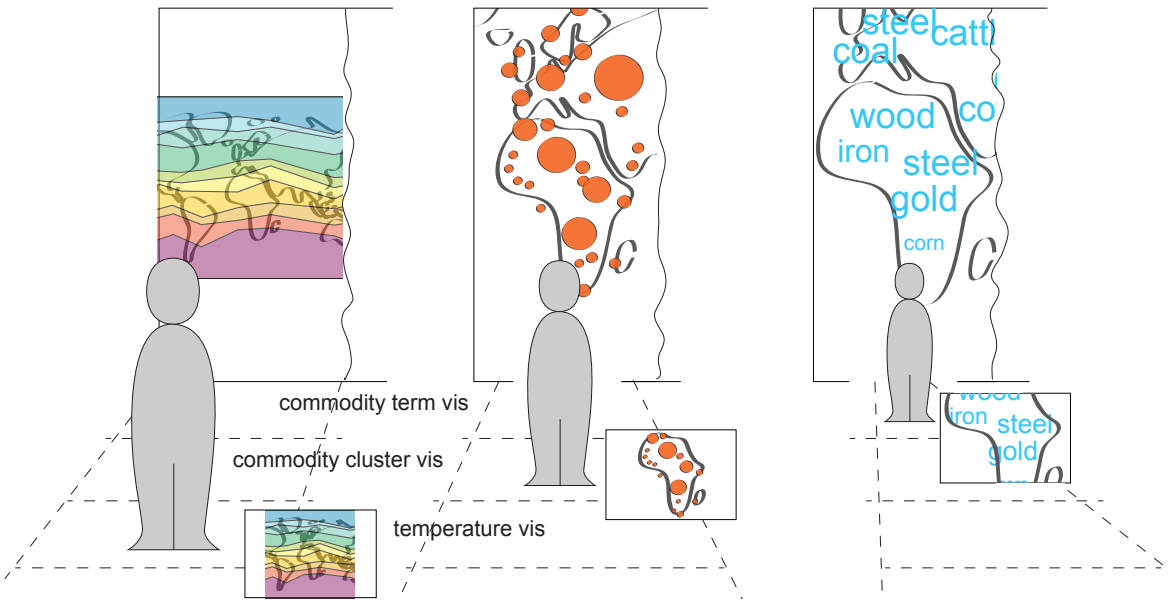


Figure 6.5: *Vis-Active* Display with *Amplified Zoom*: the distance to the display determines (1) the type of visualisation and (2) the amount of context/magnification level of information. (Figure by Uta Hinrichs)

observe the visualisations that their collaborators are working on in their periphery. This may inspire further explorations of their own visualisation. However, the constant viewing angle has the limitation that comparisons of size between data items in the different visualisations are difficult if

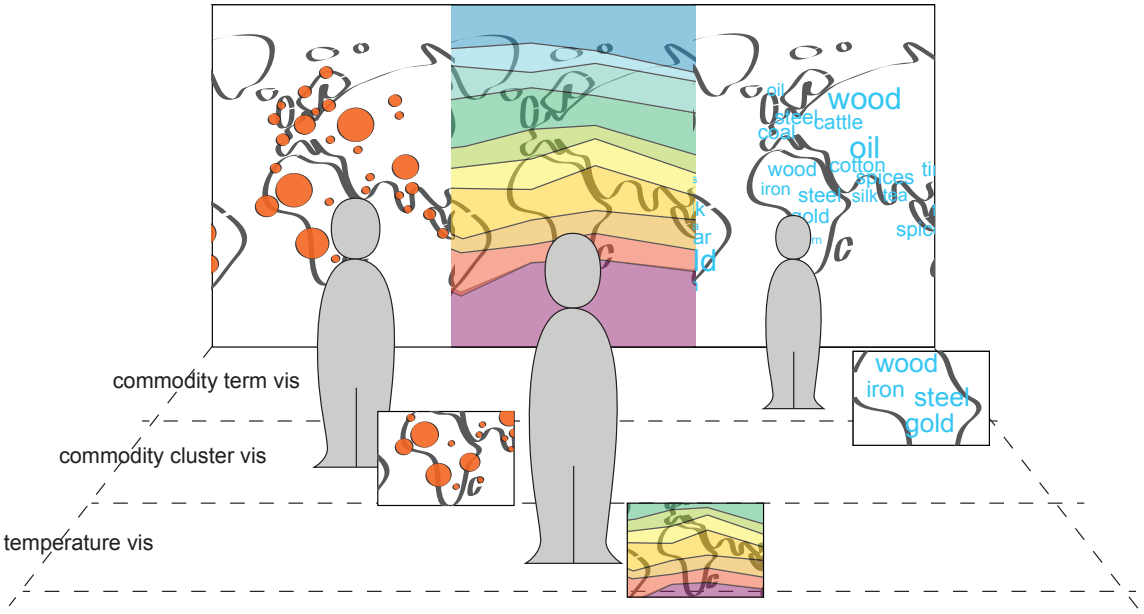


Figure 6.6: *Vis-Active* Display with *Physical Zoom*: multi-user scenario. (Figure by Uta Hinrichs)

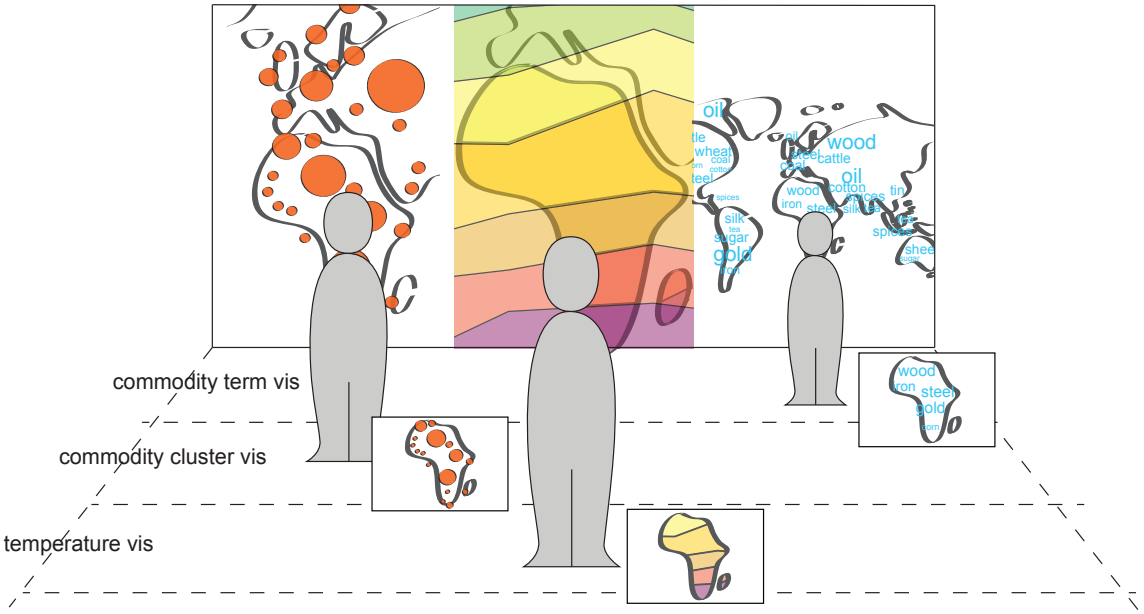


Figure 6.7: *Vis-Active* Display with *Constant Zoom*: multi-user scenario. (Figure by Uta Hinrichs)

group members stand in different zones since the viewing angle remains constant with changing distances to the display. However, as group members start to collaborate in a more closely coupled way on two different visualisations, for instance to actively compare trends within different data, it is likely that they will choose to stand in horizontal proximity, according to previous studies [HK+05].

Scenario 3: Vis-Active with Amplified Zoom Figure 6.5 shows a version of the *Vis-Active* scenario that is effectively the inverse of Scenario 2: more context information is shown from afar, while content becomes magnified as the person moves closer to the display. Note that in both variations, the type of visualisation is also changed according to the distance to the display, as described in Scenario 2. Also, while the magnification behaviour is inverted, the advantages and disadvantages when used by multiple users are likely to be similar as with *Constant Zoom* in Scenario 2.

Scenario 4: Detail-Active Display with Constant Zoom In Scenario 4, the *Detail-Active* Display (see Figure 6.8), the level of data detail within the same visualisation is changed as people move toward and away from the display. The assumptions is that showing more detail of the data can be helpful when people are close to the display for perceptual and interaction reasons. Standing close to the display makes it possible to *perceive* more subtle nuances and distinguishable features within the data that may not even be visible from further away, even if they would be represented. Furthermore, people may want to engage in more elaborate active explorations (e.g. via direct-touch), in which case it makes sense to show more data and, therefore, provide a more fine-grained visualisation of the dataset. Only one variation of the *Detail-Active* Display is depicted. It uses *Constant Zoom* and directly corresponds to the concept shown in Figure 6.9. Further variations using *Physical Zoom* and *Amplified Zoom* are also possible and are analogous to Scenarios 2 and 3, respectively.

In the multi-user *Detail-Active* Display with *Constant Zoom* scenario, group members could, again, focus on different (previously chosen) visualisations, which will remain the same as they move back and forth in front of the display (see Figure 6.9). The level of detail for each individual group member and the viewport on the visualisation change as a group member's distance to the display is altered. In a collaborative scenario, providing different levels of detail along with different types of visualisations can be beneficial: similar to the other scenarios described, group members can work in parallel to explore different perspectives of the data. In more closely-coupled collaborative phases which may be about discussing particular patterns or discoveries, it can be beneficial to have different levels of detail on the data available and blend visualisations as described in Scenario 1.

Scenario 5: Complex Combinations The *Vis-Active* and *Detail-Active* scenarios can be seen as two special cases of a single hierarchical structure. Each detail layer is essentially the same visualisation with a different amount of detail visible. Therefore, each visualisation layer in the *Vis-Active* scenario can be defined as a visualisation layer containing only a single detail layer. This means that the scenarios can be united by defining the visualisation set as a set of one or more visualisation layers, each of which contains one or more detail layers. Figure 6.10 is an example design of a complex *Vis-Detail-Active* visualisation set. In even more complex cases with multiple datasets, there would be one visualisation set per dataset.

6.4 Collaboration around Large Displays

Previous work has identified a number of general factors that influence small-group collaboration, including spatial considerations regarding the position of people and objects to each other [Hal66; SSI04] along with explicit and implicit mechanisms to foster communication and coordination of the interplay between individual and collaborative activities (e.g. [GG00; KC+04; Tan91]).

In parallel, a large body of work has explored how to support collaborative scenarios around large wall-sized displays in informal and formal office settings (e.g. [HL+10; MI+99; PM+93; SG+99]), communal spaces (e.g. [BI+04; CN+03]), and public environments (e.g. [JM+10; PK+08; PR+03; VB04]). However, little prior work has explored the role of spatial relationships for shared and collaborative vertical display systems [BMG10; HK+05; VB04]. This section outlines findings from this previous

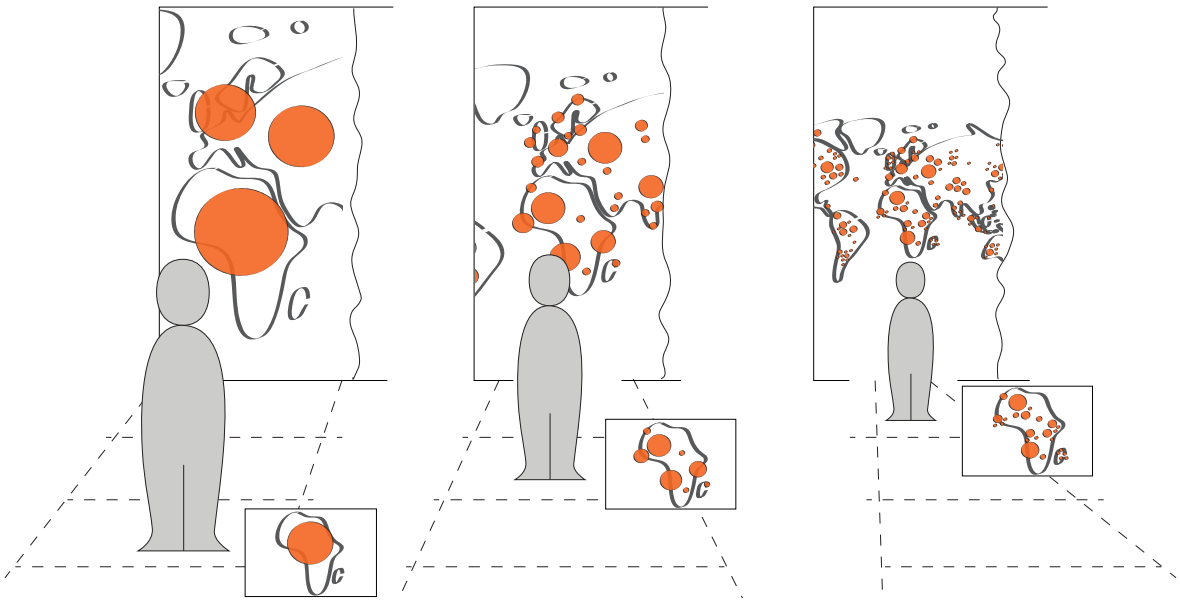


Figure 6.8: *Detail-Active* Display with *Constant Zoom*: proximity to the display determines the level of data granularity. (Figure by Uta Hinrichs)

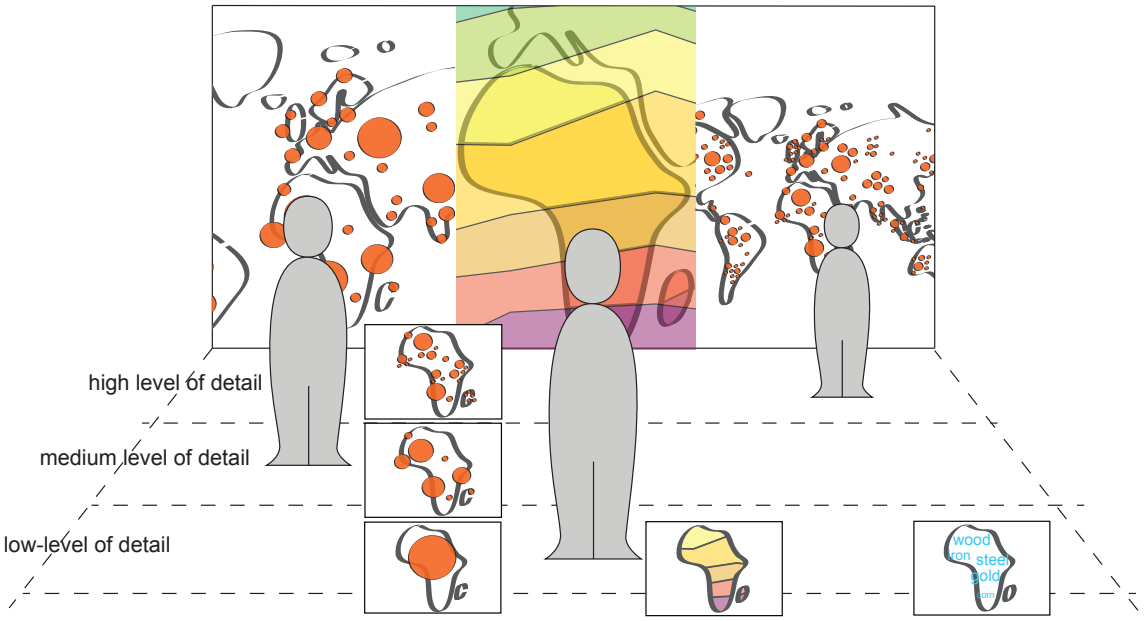


Figure 6.9: *Detail-Active* Display with *Constant Zoom*: multi-user scenario. (Figure by Uta Hinrichs)

research, on exploiting spatial relationships to support co-located small-group collaboration around large vertical displays.

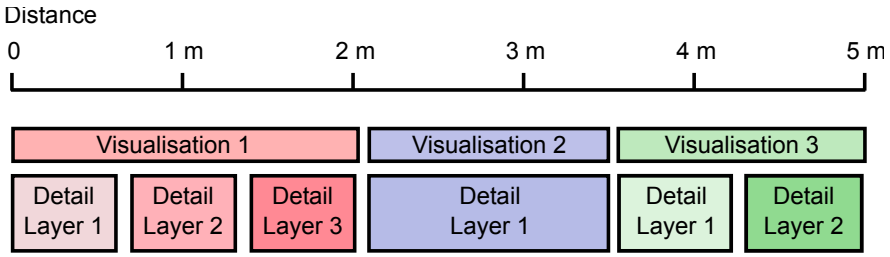


Figure 6.10: An illustration of a complex visualisation set consisting of three visualisations with each visualisation using a different amount of detail layers.

6.4.1 In-Parallel and Collaborative Work Phases

Previous work in the area of CSCW suggests that collaborative activities are typically defined by phases of work in-parallel and more closely-coupled collaboration [SGM03; TT+06; Tan91]. Enabling the creation of personal workspaces where individual activities can be carried out, as well as shared spaces in which collaborative activities can take place, is therefore important — not only for tabletop systems [SGM03; SSI04; TT+06], but also for vertical displays [HL+10; JM+10; VB04]. Since transitions between individual and collaborative work phases are fluid, personal and collaborative spaces should be flexibly adjustable without much effort [SSI04]. This notion of establishing flexible personal and shared workspaces is considered in the context of shared, spatial interactions in front of large vertical displays. In particular, we explore mechanisms for *workspace negotiation* that are driven by information about an individual's distance to the display, as well as their distance to other group members.

6.4.2 User Arrangement

Large displays allow considerable flexibility when it comes to the arrangement of group members around or in front of the display. When collaborating around a wall-sized display there may be phases where people directly manipulate information on the display and phases where people step back in order to gain an overview of the information gathered or to discuss further steps. Prior work on collaborative scenarios around wall-sized displays has found that the arrangement of people in front of the display has implications for the character of collaborative activities [HK+05; MSI02; RL04].

Distance to the Display Previous studies have found that the distance to the display can influence the dynamics of collaboration and establishes particular roles among group members. Rogers and Lindsey [RL04] found that small groups collaborating around a wall display elected a person "in charge" of coordinating group activities. This person would typically be positioned close to the display while the other group members would take a more passive role and gather around the informally designated person-in-charge. These findings were confirmed by Hawkey et al. [HK+05], who investigated the influence of distance on co-located collaboration in a more systematic way. Even in public settings where activities evolved in an ad-hoc and spontaneous way with people interacting in front of a multi-touch wall display, groups were observed establishing roles with one person being "in charge" of driving the interaction while the other group members formed a passive audience, even if simultaneous interactions were supported [PK+08].

While social connotations may lead group members to take on somewhat unequal roles as part of their collaborative activities, these findings may be influenced by the interaction techniques provided to manipulate information on the display. As Hawkey et al. [HK+05] report, participants preferred

direct-touch interaction over more cumbersome and indirect interactions with the display from afar. The work in this chapter explores ways to share information on large wall displays through body movement. This may lead to different, more equalised group dynamics.

Type of Information in the Context of Distance. The distance to the display influences the view of the presented information. With groups collaborating on a map interface, Hawkey et al. [HK+05] found that some groups actually preferred the option of collaborating on their tasks from a larger distance to the display. This suggests that collaboration on certain types of (visual) information may be more comfortable and/or effective from a larger distance while other types of information may be easier to explore from a closer distance. Parts of the work in this chapter explore how moving back and forth in front of the display can control the representation of information.

Distance Between Group Members. Prior work has investigated the influence of distance between group members on collaborative activities. Hawkey et al. [HK+05] compared work strategies of pairs working (a) directly in front of a vertical display, (b) both at a distance from the display, and (c) with one group member in front of the display and one group member at a distance from the display. They found that participants felt comparatively comfortable working both close or both at a distance from the display. However, collaboration suffered if group members were not at the same distance to the display. In this condition, communication and coordination of activities were severely hampered.

Findings from observational studies of large public wall-sized displays has revealed that interacting at different vertical distances seems to indicate independent interactions, rather than collaborative activities [MW+12]. This chapter expands on these findings by exploring mechanisms to establish shared and personal workspaces based on people's (vertical and horizontal) distance to each other.

6.5 Design Considerations

Exploration, implementation and testing of the scenarios described above, especially when coupled with the results of existing research, reveal a number of factors other than content and view manipulation that should be considered when designing interaction techniques that manipulate visual content, especially in collaborative settings. This section summarises the findings and observations as recommendations and considerations that designers of future systems should take into account.

6.5.1 General Considerations

Even though wall displays offer ample display space, when sharing the space with other users, collaboration scenarios may need to be accommodated or conflicts may arise. Three general design considerations for building wall display shared spaces are presented.

Effective Use of Display Space The first design consideration concerns the effective use of space. One of the design goals in the Flatland project was to always allow creation of additional white space on the board as long as the visibility of existing content was maintained [MI+99]. The principle of effective use of display space goes in the opposite direction. Unlike in the open-ended use scenarios with a whiteboard, the amount of content displayed on the wall display is always limited and finite. Allocating the maximum possible space for existing display content allows for more efficient use of the available display space.

Effective Information Density Closely related to the previous consideration is choosing the appropriate amount of information to display within the allocated display space. The first intuition seems to be that more information is generally beneficial. However, due to physical constraints (e.g. visual acuity of the user), cognitive constraints (e.g. prevention of information overload) and environmental constraints (e.g. the pixel resolution and density of the display), the designers of interactive visualisation spaces should carefully balance the amount of information shown at each distance level from the display.

Fluid Transitions The system should provide fluid transition between states. Rather than abruptly changing the state of the system, the system first indicates that change is imminent in one of two ways - using a resistance band or a balance transition. They both apply the fluid transition principle in different ways. The resistance band indicator can be best described using an analogy of two magnets stuck together. When applying force to separate the magnets, the magnets stay locked together until the force trying to separate them reaches a threshold at which point the magnets separate. This means that when a boundary for a transition is reached, the boundary is 'bent' in an attempt to maintain the current state until the intention of the user to change the state is made clear by reaching the end of the resistance band. This indicator allows fluid, yet stable transitions between states.

The balance transition is based on the principle of communicating vessels. The principle states that a liquid in a set of connected containers will always balance out at the same level when it settles. In our system, when the user approaches the state boundary of one state, the other state will start seeping into the current state. At the boundary, the view opacity of the two states will be equal. As the user enters the other state further, the visibility of the past state lowers progressively. This transition gives the user a clear indication of the state of the system and the imminent change in a flexible and fluid manner.

6.5.2 Strategies for Space Negotiation

In applying the design considerations, five distinct strategies for negotiating display space on a large shared display are proposed. Every strategy can be applied on its own but most of the strategies can be applied in concert with one another. Each of the five strategies applies the design considerations in a different way.

Blend Blend is a strategy for creating a shared collaboration space. It makes use of the balance transition. When two users approach each other to share space, the content on each user's space starts to seep into the other's space. After a pre-defined time interval, the two content sets will be shown in a balanced blend and the individual spaces become a single shared space. This gives the users an opportunity to disengage before the transition completes should they not want to collaborate after all. This strategy is most likely to be useful when the content representations are visually compatible (such as the map based views used in the example scenarios). However, where the visual representations have little in common, other strategies, such as Shift or Avoid will likely be better suited.

Merge Merge is also a strategy for creating a shared space. However, unlike Blend, it uses the resistance band transition. When two users approach each other to share their spaces, no visible action occurs at first. As their relative distance shortens, they eventually reach the threshold of the resistance band, at which point their individual spaces merge together into a single space. Again, the resistance band offers the opportunity to stop the transition and/or take and alternative action.

Merge shares the same weakness as Blend in that visually diverse content representations are unlikely to merge well without manipulations to the content.

Shift Shift is a strategy for accommodating parallel interactions by maintaining separate personal views of users. Instead of creating a shared visualisation space, Shift aims to resolve potential conflict by shifting the personal spaces intact either vertically or horizontally. This strategy is similar to a combination of the move and bump mechanisms described by Mynatt et al. [MI+99]. No overlap of the personal spaces is allowed. When conflicting claims for a particular region of the display arise, the balance transition is used to shift the personal spaces in a fluid manner. Shift may offer a direct replacement alternative to Blend and Merge in cases where the two strategies would not produce usable shared spaces.

Avoid Avoid is also a strategy for avoiding conflict and maintaining separate personal view. However, unlike Shift, which allows the views of other users to be shifted, Avoid always respects the position of the other users' spaces, only moving the active user's space to an unoccupied region of space. Overlapping the personal spaces is also not allowed with Avoid. If the path to the unoccupied space is obstructed by another user's space, the resistance band transition is used to bounce the user's space across.

Expand Expand is a strategy, which directly stems from the aim to effectively use the available display space. Unless forced to keep its size by other constraints, a user's personal space will always expand to the available display space not occupied by other users.

Variants of these strategies are already in use in other systems, demonstrating their validity. For example *ProximityTable* utilises a version of the Merge strategy within its tabletop interactions [Aln15]. However, the performance and detailed interactive characteristics of the strategies have not been experimentally confirmed yet, and so they remain a target for possible future research. One of the quickest ways to develop an experimental setting for such evaluation would be to use the design tool, which is presented in the next section.

6.6 Toolkit Overview

All the scenarios presented in Section 6.3 were implemented using a prototyping tool presented here. The tool is part of the SpiderEyes prototyping toolkit, which also includes a tracking system. The main features of the toolkit include:

Multi-User Tracking: The toolkit tracks up to four users in real-time.

Separates Foreground and Background Activity: The toolkit uses computer vision to track users' eyes and uses this information to separate users actively engaging with the system from users in the background attending to other activities or just passing by.

Markerless Tracking: The system does not use any markers to track users and does not require any calibration for users to be tracked.

Easy Setup: The toolkit only relies on a single depth camera (e.g. a Kinect) and a high-resolution RGB camera, making it easy to set up and deploy in a variety of environments.

Programming Language Independent: The tracking system communicates its results in a programming language independent format, which allows designers to use a programming language of their choice.

Distributed Deployment: The tracking system can be deployed on a different computer than the application that uses it. This allows developers more flexibility and independence from specific deployment environments.

Prototyping Tool: The toolkit contains a web-based tool for designing visualisation sets. It allows designers to make their existing or novel visualisations spatially aware.

The toolkit is realised via two components. The first component is a multi-user tracking system. Developers can use this component to obtain real-time information about multiple users' positions and attention-aware statuses in relation to a large wall-sized display. The second component is a design tool that enables toolkit users to design attention-aware visualisation applications.

6.6.1 Toolkit Component 1: Tracking System

Developing markerless multi-user spatial tracking systems is a non-trivial task. To enable designers to easily create new spatially-aware interfaces, a flexible toolkit that provides easy-to-use programming abstractions was developed. While the bulk of the system is written in C++, the tracking system runs either a TCP/IP or a WebSockets server, which enables both native applications and browser-based applications to use the data.

Currently, the tracking system is configured by defining several simple parameters about the environment in a text file. Designers can choose from several tracking algorithms (Computer Vision (CV) only, multi-user Kinect-CV fusion, or single-user Kinect-CV fusion) as well as configuring the data output (TCP/IP server, WebSockets server, or local logging). Once the system is running, the server sends the tracking results to all connected clients as a valid JSON object. See Figure 6.11 for an example of a data point.

The tracking system leverages algorithms that were developed as part of the tracking method efforts described in detail in Chapter 4. The specifics of the algorithms used in this toolkit are detailed in Sections 4.3 and 4.4 of the chapter.

```
{
"timestamp": 0003030300,
"users": [
   {"userid": 1,
    "position": {"x": 1234, "y": 1750, "z": 1063},
    "orientation": {"x": 0, "y": 0, "z": -1},
    "confidence": 1},
   {"userid": 1,
    "position": {"x": -570, "y": 1640, "z": 1534},
    "orientation": {"x": 0, "y": 0, "z": 0},
    "confidence": 0}
]
}
```

Figure 6.11: Example JSON object sent by the tracking system. The time-stamp is in milliseconds since the system has been initialised; position is in millimeters; orientation is reported as a unit vector. Both position and orientation are in sensor coordinates. Confidence is in the interval 0-1 (values reported by the system: 1 = full detection: eye-pair and individual eyes are detected; 0.8 = partial detection: only the eye-pair is detected; 0 = no detection).

```
var entity_size_type = "angle";
var layer_zoom_type = "constant";
var layer_angle = 20.0; //in degrees
var amplified_midpoint_distance = 1500; //mm
var zoom_amplification = 0.5; //1.0 = neutral
var group_distance = 400; //mm

function generateVisualisationSet(uid) {
   var words = new detailLayer("w.svg", 0, 5000); //detailLayer(url, start_dist, end_dist)
   var heat = new detailLayer("h.svg", 0, 5000);
   var clusters = new detailLayer("c.svg", 0, 5000);
   var default_visLayer = new visualisationLayer();
   switch(uid) {
   case 1: {
      default_visLayer.addLayer(words);
      break; }
   case 2: {
      default_visLayer.addLayer(heat);
      break; }
   case 3: {
      default_visLayer.addLayer(clusters);
      break; }
   case 4: {
      default_visLayer.addLayer(heat);
      break; }
   }
   var result = new visualisationSet();
   result.addLayer(default_visLayer);
   return result;
}
```

Figure 6.12: Code listing for the JavaScript API used for implementing the scenarios. In this code sample, the *generateVisualisationSet()* function implements the passive scenario with constant zoom and a different visualisation for each user.

6.6.2 Toolkit Component 2: Rapid Prototyping Tool

The rapid prototyping tool is web-based and written in JavaScript. The use of the prototyping tool requires setting the values of several parameters and the implementation of a single function. The following parameters can be set (Figure 6.12 provides an example implementation for the *Vis-Active* Display with *Constant Zoom*):

Viewport Sizing (*entity_size_type*): *fraction* This parameter gives each active user an equal fraction of the display space. *angle* will dynamically resize the viewports so that they occupy equal visual angles for all active users.

Layer Magnification (*layer_zoom_type*): This parameter defines which of the magnification methods described in the scenarios section should be used for the visualisation layers. Possible values are: *physical, constant, amplified, inverse*. For *constant*, an additional parameter that defines the desired visual angle must be set (*layer_angle*). For *amplified* and *inverse*, the amplification ratio (*zoom_amplification*) and neutral point (*amplified_midpoint_distance*) need to be set.

Default Visualisations (*generateVisualisationSet(uid)*): This function allows the designer to define the visualisations and their distance boundaries, for each user. Each detail layer is defined by the

url to its content (which can be an image or a URL to a webpage) and the *start* and *end* distance boundaries for its visibility.

Grouping Distance (*group_distance*): This parameter defines the maximum distance between a pair of users for them to be considered a group by the system.

The design tool automatically manages the creation of the application itself. In addition, the tool automatically distinguishes active users and people passing by in the background and foreground based on whether their visual attention is on the display or not and only displays visualisations for active users.

6.6.3 Exploiting Spatial Interactions Beyond the Outlined Scenarios

In order to demonstrate the power of the prototyping tool beyond visual content manipulation scenarios, a spatially-aware interactive visualisation was implemented. The rapid application creation was utilised only marginally in order to position the interactive visualisation. Since the prototyping tool also has the potential to expose the tracking data to any visualisations utilising the toolkit, it became possible to augment an existing visualisation to add interactivity.

In addition to the demonstration of the flexibility of the toolkit, the purpose was to evaluate the difficulty of interfacing the toolkit with an established information visualisation framework (in this case the D3 visualisation library). For this, the "Wealth and Health of Nations" visualisation was selected as an example[1]. This example visualises a complex, high-dimensional dataset (country, per-capita income, life expectancy, population size, and time).

In the implementation, the lateral movement along the horizontal axis was mapped to the temporal dimension of the dataset. Stepping from one side to another in front of the wall-sized display changed the displayed data to a specific year. This augmented interactive visualisation can be seen in Figure 6.13, Figure 6.14, and in a supplemental video figure for this chapter.

Many alternative designs are also possible. The forward and backward physical movements can be mapped to a scale/zoom mechanism, such as *Constant Zoom*. While *Constant Zoom* is used, the lateral movements along the horizontal axis may also be mapped to a translation function, moving the position of the visualisation on the display so that it is always centred in front of the user.

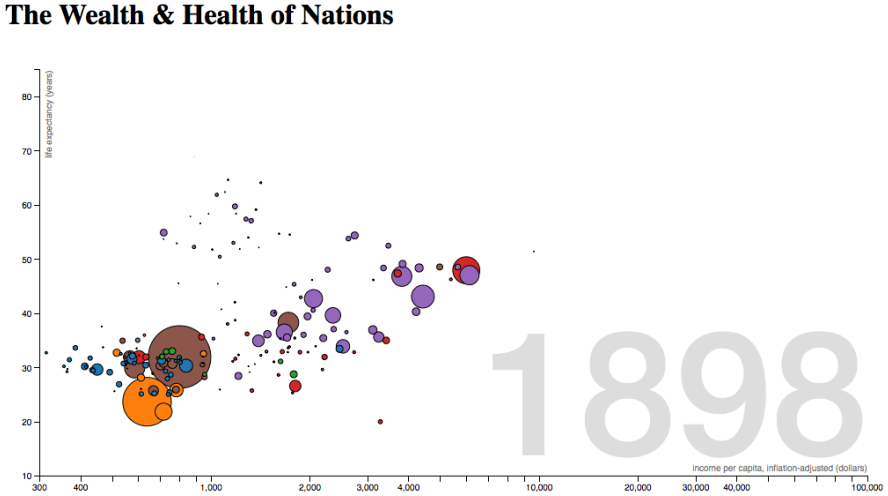


Figure 6.13: The "Wealth and Health of Nations" D3 visualisation used as an example.

[1] from `http://bost.ocks.org/mike/nations/`

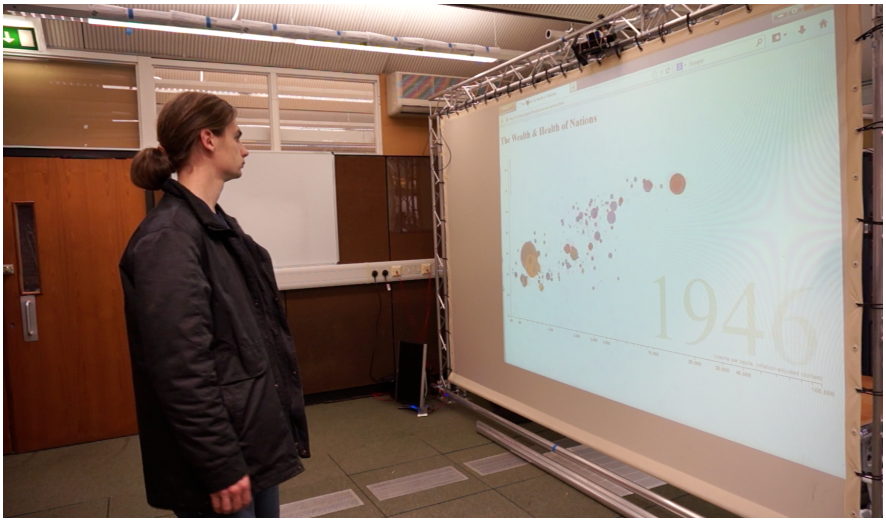


Figure 6.14: The "Wealth and Health of Nations" D3 visualisation explored by a user using lateral movement to literally step through time.

	Distance					
Name	AA	AO	AE	OO	OE	EE
SpiderEyes	C	C				

Table 6.2: Relationship - Distance

Relationship legend: A - Actor, O - Object, E - Environment; e.g. AO — Actor-Object
Value legend: C - Continuous adaptation; D - Discrete, zone or threshold based adaptation; P - Binary adaptation based on presence/absence

	Position					
Name	AA	AO	AE	OO	OE	EE
SpiderEyes	(C)	C	(C)			

Table 6.3: Relationships - Position

Relationship legend: A - Actor, O - Object, E - Environment; e.g. AE — Actor-Environment
Value legend: C - Continuous adaptation; D - Discrete, zone or threshold based adaptation; P - Binary adaptation based on presence/absence. Values in brackets are not used in the example applications provided but are available as part of the *SpiderEyes* toolkit.

6.7 SpiderEyes and the IRE Model

This section provides another example analysis of a system using the IRE model. To concisely summarise the system, *SpiderEyes* is an application system and part of a toolkit for developing spatially aware visual interfaces. By default, it comes with several example spatial techniques for manipulating visual on-display content. These techniques form the basis of this analysis. The size of the displayed content can be changed based on the distance to the display (magnification). The detail granularity can be altered (semantic zoom) or changed altogether through the use of different visual representations or the use of multiple datasets. The position along the display can determine the positioning of a person's viewport in multi-user scenarios. Alternatively, it can be used as a control

	Orientation								
Name	AA	AO	AE	OO	OA	OE	EE	EA	EO
SpiderEyes	(C)	C			(C)				

Table 6.4: Relationships - Orientation

Relationship legend: A - Actor, O - Object, E - Environment; e.g. AO — Actor-Object
Value legend: C - Continuous adaptation; D - Discrete, zone or threshold based adaptation. Values in brackets are not used in the example applications provided but are available as part of the toolkit.

	Range			
	intimate	personal	social	public
Name	(<0.6m)	(<1.5m)	(<5m)	(>5m)
SpiderEyes		x	x	

Table 6.5: Interaction - Range

Value legend: x - System was described or demonstrated in use within this distance range.

	Cardinality					
Name	Actor				Object	Environment
SpiderEyes	1	1*	M	M*	1	

Table 6.6: Interaction - Cardinality

	Mode						
Name	Actor			Object			Environment
SpiderEyes	Sp	Vis	Sym	Sp	Vis	Sym	

Table 6.7: Interaction - Mode

Value legend: Sp - Spatial, Vis - Visual, Int - Intent, Sym - Symbolic, Ac - Acoustic

	Intentionality			
Name	Actor		Object	Environment
SpiderEyes	E+	I+	E	

Table 6.8: Interaction - Intentionality

Value legend: E - Explicit, E+ - Explicit (with possible misclassifications), I - Implicit, I+ - Implicit (with possible misclassifications), A - Ambient

	Intensity				
Name	Actor	Object			Environment
SpiderEyes	N	U	S	N	

Table 6.9: Interaction - Intensity

Value legend: U - Unnoticeable, S - Subtle, N - Neutral, I - Intrusive, D - Disruptive

mechanism. People's orientation towards the display can be used to filter out passers by. Distance to other users is used for determining grouping.

6.7.1 Entities and Relationships

There are two types of entities in *SpiderEyes*. The large display is the main interactive Object, while the people interacting with the display are in control of the interaction as Actors. The environment is not used as an interactive entity.

All three kinds of spatial relationships are utilised within *SpiderEyes*. Starting with distance, the Actor-Object and Object-Object distances are used. The Actor-Actor distance is used as a mechanism to form groups when interacting with the system. The Actor-Object distance is employed as a control mechanism for changing content on the large display. This can be either a change in magnification, detail granularity or switching between visualisations and datasets. The distance relationships are summarised in Table 6.2.

For position and orientation relationships there is an interesting situation where the *SpiderEyes* toolkit enables greater use of those relationships than what is used within the exploration of gaps in the Content Manipulation Matrix. The visualisation scenario where the position along the large display determines the point in time that the visualisation shows fall into the Actor-Object position relationship. However, since the toolkit exposes the positions of all the tracked persons as a position within the interactive space, the positional relationships between them (Actor-Actor position) or specific positions within the interactive environment (Actor-Environment position) could potentially be used for interactive purposes. Table 6.2 summarises the values for position.

The same applies to orientation relationships. Only Actor-Object orientation is actively used during the explorations of the Content Manipulation Matrix as a filtering mechanism to distinguish between passers-by and active users of the system based on their orientation towards the large display. However, Object-Actor orientation and Actor-Actor orientation relationships can be derived from the information exposed by the toolkit, potentially enabling other interaction techniques. Since those relationships are not actively used within the explored scenarios but the potential for use is present, the values in the respective tables are in brackets to denote this.

With all spatial relationships used within *SpiderEyes*, the relationships are primarily continuous. However, it is possible for them to be used in their discrete or presence/absence forms too.

6.7.2 Interaction

Moving to interaction measures and starting with range, the *SpiderEyes* system enables interactions at the personal and social ranges. Table 6.5 summarises the values. The ranges are mostly dictated by the sensing capabilities of the system and the display setup. While the combined RGB+depth camera tracking system can track entities within the intimate range in relation to the sensor, the field of view of the cameras means that only a portion of the space in front of the large display is covered within the range. By the time the personal range is reached, the entire width of the display is covered. Both the size of the room as well as the tracking system limit the maximum interaction range to approximately 4.5-5 metres, which means that the social range is covered but not the public range.

SpiderEyes is very easy to analyse with regards to the cardinality measure. The only Object in the system is the large display, and there can only be one, which means that only `1` is possible as a cardinality value for Objects. For Actors, any value is possible. This is demonstrated in the supplementary video for this chapter and partially in Figure 6.1. The only cardinality value not demonstrated visually is `M*`, but the system has been used by two groups of two as part of system tests.

Entities in *SpiderEyes* make use of the Spatial, Visual and Symbolic modes. The Spatial mode is demonstrated through the spatial interaction techniques, the Visual mode is shown through the use of visual information on the large display and the entities are classified as using the Symbolic mode

because some of the visualisations utilise textual linguistic information. The values are summarised in Table 6.7.

Intentionality values for *SpiderEyes* are shown in Table 6.8. The Object of the interactions is the display and all display output is explicit within the system. Actions by Actors fall into the explicit and implicit categories. In both cases, however, the system makes assumptions about the actions, which means that it is possible for a misclassification of the action to occur. The explicit actions include using distance and position changes to control the visualisation. The inherent assumption is that the Actor's movement is always intended to change the visualisation, even in cases, where they may simply be negotiating spatial positioning in relation to other Actors. Because of this assumption, the value is *explicit+*. Orientation of an Actor towards the display is interpreted as an implicit intention to interact, again with the assumption that this is always the case. This means that the actions are classified as *implicit+*.

Most actions that occur within the system are classified as having *neutral* intensity. The exceptions are some of the magnification techniques (specifically constant zoom) used on the display, which can be quite *subtle*. The *subtle* and *unnoticieable* values additionally come from interactions with passers-by where by definition, until the passer-by actually looks at the display, they will not be able to see the content on the display (or only in their peripheral vision). The values for intensity are summarised in Table 6.9.

To conclude, in addition to providing another example analysis using the IRE model, this section highlights that a distinction may need to be made between the interactive potential of a system and the relationships and other interactive properties that the system demonstrably makes use of. The analysis also highlights the hierarchy of spatial relationship forms. The continuous relationship form is the most general relationship form, with the discrete relationship being a more constrained category within that. The presence/absence relationship form is a special case of the discrete relationship form.

6.8 Summary and Conclusions

As seen in Chapter 3, there are a number of research systems, which make use of spatial relationships (most commonly distance) to visually manipulate content on large displays. This chapter presented an systematic exploration of content manipulation methods based on existing literature. This exploration resulted in the introduction of the Content Manipulation Matrix, which enabled a gap analysis to be performed. This analysis showed that most existing systems have not explored a large portion of the parameter combinations.

In order to demonstrate the potential of the previously unexplored content manipulation parameter combinations, five example scenarios were devised using an example use case with knowledge workers exploring visualisations of a multi-faceted dataset. As the exploration, implementation and informal testing of the scenarios revealed a number of concerns future researchers and designers may face, a set of design considerations was devised. These design considerations, grounded in research literature address some of the concerns and propose approaches to mitigate them.

Lastly, this chapter introduces a prototyping tool, which enables researchers and designers to rapidly prototype content manipulation techniques based on the Content Manipulation Matrix. The details of the tracking algorithms are described in detail in Chapter 4. The capability of the prototyping tool beyond the explored scenarios was demonstrated by augmenting an existing visualisation to use the position of its user.

The presented research clearly demonstrates both the research opportunities offered by the Content Manipulation Matrix as well as the capabilities of the prototyping tools.

Chapter 7

Case Study 3 - Techniques for Inattention in Multi-Display Environments

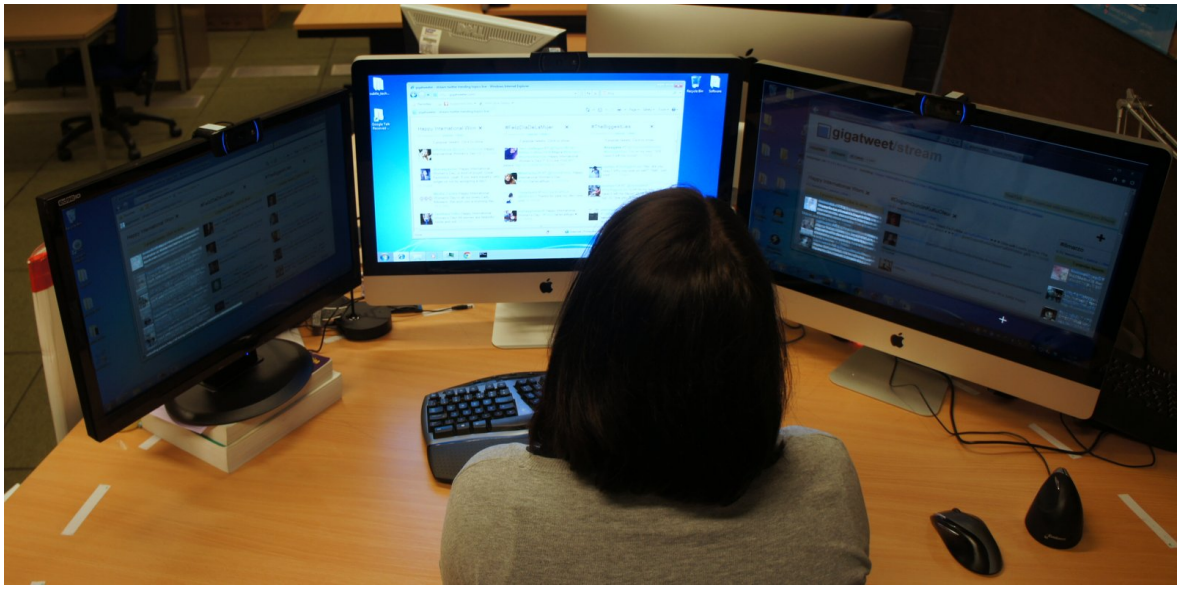


Figure 7.1: An illustration of one of the techniques for visualising changes on unattended displays in action on the two peripheral displays. The PixMap technique highlights any pixels that change as a temporal heatmap.

Modern computer workstation setups regularly include multiple displays in various configurations. With such multi-display setups it is possible to have more display real-estate available than a person is able to comfortably attend to. While the benefits of large or multi-display setups have been demonstrated in several studies (e.g. [BG+02; BB09; CS+03; Sim01]), it has also been suggested that this increase in display space will lead to usability problems [CS+03], window management difficulties [BB09] and issues related to information overload [Int01].

Another potential issue is change blindness: people's inability to detect significant visual display changes when there is a disruption in continuity such as a brief flicker or a shift in visual focus. However, the effects of change blindness in multi-display environments have not been extensively studied in the literature. In one study, DiVita and colleagues [DO+04] report that change blindness

was a significant factor for operators managing critical events using multi-display command and control systems with unattended displays.

In general, the increased display real-estate afforded by multi-display setups means that people are unable to attend to all of it at once. In particular, this point is reached when the total display area is so large that it does not fit within the person's field of vision. In this case, the person has to substantially turn their head to see different parts of the display environment. This situation eventually arises when the number of displays or the distance between them increases. For example, it is likely to occur when people are working with three displays aligned bezel to bezel (e.g. Figure 7.1). When a person is only able to observe part of the multi-display environment, changes occurring on the unattended displays are difficult to track.

Attentive user interfaces [VD+02; VS+06] and distraction reduction techniques [SB05a] have been extensively studied previously using a variety of context-sensitive eye- and gaze-tracking technologies. In contrast, *inattention*, and specifically technologies that track visual change on unattended displays, has received considerably less attention.

Most closely, Bezerianos et al. [BDB06] explored tracking of pixel changes in application windows during periods of occlusion in sandboxed environments on a single desktop or large display. In 2005, Ashdown and Sato presented a multi-display system, which uses head tracking to reposition the mouse pointer to the display the person attends to [AS05]. Another example is the work by Kern et al. [KKS10] on Gazemarks: a visual placeholder that indicates the person's last fixation on a display. Kern et al. [KKS10] reported that the Gazemarks technique resulted in faster completion times in a map navigation task in a vehicle driving scenario. Finally, Bi and Balakrishnan [BB09] suggest further investigation of awareness of peripheral applications with large, single and dual desktop setups as a fruitful avenue for future research. However, research on change blindness in multi-display environments has shown that making people aware of changes in unattended displays is a significant challenge [DO+04].

This chapter introduces *DiffDisplays*, a system that enables tracking and visualisation of visual changed on unattended displays. Several additional contributions are made in the remainder of this chapter. The first contribution is four visualisation techniques for assisting people in perceiving and tracking display changes in multi-display environments. Second, the results of a five-day case study in which a working professional used *DiffDisplays* for 39.25 hours as part of his regular work activities. Finally, this chapter includes a discussion of the challenges in evaluating subtle intelligent visualisation techniques.

7.1 Approach

Four subtle gaze-dependent techniques for visualising display changes have been designed. The techniques are called FreezeFrame, PixMap, WindowMap and Aura. One can view this contribution within the attentive user interface framework proposed by Vertegaal et. al [VS+06]. With respect to this framework, *DiffDisplays* attempts to sense and communicate changes due to *inattention*.

The techniques are designed to be used within an existing work context of a person. As a consequence, it is important that the techniques do not distract the person from their primary task. Therefore, the techniques are designed as *calm technologies*, as proposed by Weiser and Brown [WB95]. A calm technology *"moves easily from the periphery of our attention, to the center, and back"* where *"the periphery is informing without overburdening"* [WB95].

To be able to implement and deploy the non-intrusive visualisation techniques in an actual work environment, a tracking system has been developed. The tracking system is able to determine which display the tracked person is attending to and, by extension, the displays that are unattended. Importantly, the system is markerless and only relies on off-the-shelf web cameras for detecting a person's orientation towards a display. This allows the system to be readily deployed in existing

workstation setups. The details of implementation of the tracking system as well as evaluation of the system as an orientation detector can be found in Section 4.3.6 in Chapter 4.

For the purpose of the study, the tracking system's ability to determine whether a person is oriented in the direction of a particular display is used as a proxy for the approximate direction of visual focus or gaze. While the tracking system is unable to provide a precise direction of gaze, the display-level granularity is sufficient for the purposes of determining the attended and unattended displays. While finer granularity is desirable for other scenarios, in this case the ability to deploy the tracking system without a significant alteration of the working environment is more important.

As a subtle interface, the prototype system does not naturally lend itself to a traditional evaluation approach, such as a short controlled experiment. Therefore, a qualitative five day study was carried out instead to better understand the finer points and nuances of using visualisations as part of an ordinary multi-display environment. A working professional was recruited as a participant for the study. The participant used *DiffDisplays* during an entire workweek as part of his regular work activities. 39.25 hours of data were recorded in total. The results show that three out of four techniques highlighted visual changes in a subtle and non-intrusive manner. Additionally, the techniques reduced distractions and the participant found them useful when working in his regular multi-display setup.

In the study, distraction reduction was one of the main aims and benefits of using the techniques. In fact, the participant reported that before taking part in the study, he was unaware of how often he was distracted until he started using the change visualisation techniques. This finding complements research into change blindness that suggests that people are not fully aware of their inability to perceive visual changes [BLA07; DO+04]. Based on the study presented here, it is possible that people may be unaware of the extent to which they get distracted by visual display changes in their environment. However, more work is required to firmly establish whether this hypothesis holds for other scenarios.

7.2 Subtle Visualisation Techniques

To assist people in noticing changes in multi-display environments, four different subtle visualisation techniques for tracking visual changes on unattended displays are introduced here. Each technique visualises display changes in a different manner. It has been established that gaze is a good approximation of the direction of attention [VS+06]. The presented techniques leverage this, in addition to using head orientation as an indicator of what is visible in a person's field of view, to trigger the visualisation techniques.

When describing the techniques, the term *frame* is used to refer to a screenshot of the contents on a display at a specific point in time. All figures demonstrating the techniques use an instant message chat (specifically Google Talk) as an example scenario. The change that occurs while a person's visual focus is elsewhere is in all cases a new instant message appearing in the chat window.

7.2.1 FreezeFrame

FreezeFrame is the simplest technique, which hides visual changes on an unattended display until the person's visual focus shifts towards it again. FreezeFrame works as follows. Consider a person working with two displays (One and Two). The person starts by attending display One. When the person switches visual focus from display One to display Two, the tracking system notices this focus shift. FreezeFrame then captures a frame of display One at the time the person shifted their visual focus to display Two. This frame is then displayed on display One as a black and white static image. No visual change is then shown on display One until the person switches their visual focus from display Two back to display One. When the person attends to display One again, the frozen black

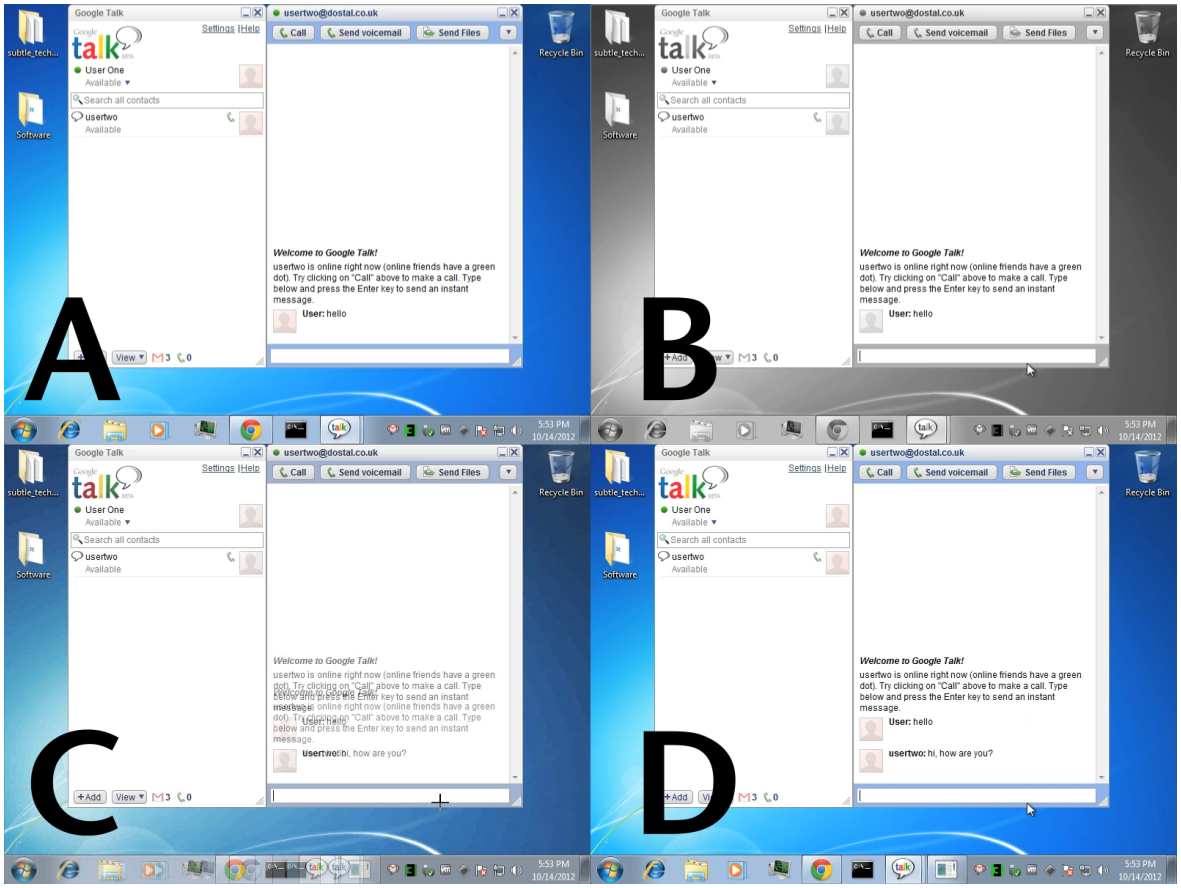
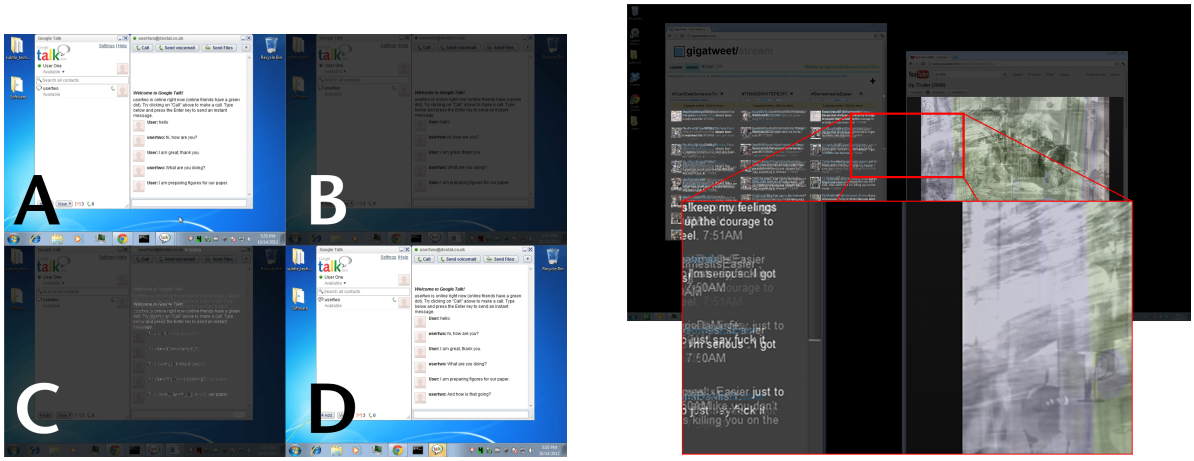


Figure 7.2: FreezeFrame reduces distractions by hiding visual change until a person shifts their visual focus back to the display. *A* shows the last frame before the person shifts visual focus away from this display. *B* shows a static frame of the unattended display before and during any visual change. *C* illustrates the dissolving of the old frame into the current display state. *D* shows the current display state.

and white frame on display One dissolves, blending smoothly into the current state of display One over the next 1–2 seconds. Figure 7.2 and Video Figure 1.1 illustrate FreezeFrame.

7.2.2 PixMap

The PixMap technique is a temporal heatmap visualisation that shows the person any changes that occurred while the display was unattended. Following the scenario previously described for FreezeFrame, when the person shifts their visual focus from display One to display Two, display One is darkened to reduce distractions. While the person's visual focus is directed elsewhere, a frame (a screenshot of the contents of display One) is captured at regular intervals, approximately six times per second. A difference in pixel values between the two most recently captured frames is computed for the entire display and the change is visualised on display One. This continues until the person's visual focus returns to display One, at which point the visualisation is dissolved into the current desktop state of display One over the next 1–2 seconds. Figure 7.3 and Video Figure 1.2 illustrate PixMap.



(a) (b)

Figure 7.3: An illustration of the PixMap technique. PixMap highlights visual changes over time on a pixel level. *A* shows the last frame before a person switches visual focus away from this display. *B* shows a frame of the unattended display before any visual change takes place. *C* illustrates how the technique visualises the change (a new instant message in the chat window) when it happens by brightening the pixels that changed in the chat window. *D* shows the current display state after the technique has been dissolved. Figure 7.3b shows a detailed example of change visualised as bright pixels.

The following formula is used to compute the value of pixels:

$$V_{new} = (V_{previous} \times Decay) + (V_{diff} \times Intensity) \tag{7.1}$$

where V_{new} is the new pixel value, $V_{previous}$ is the pixel value of the corresponding pixel in the previous iteration of the heat map. V_{diff} is the intensity of change between the last frame and the current frame measured as the difference in RGB values for the corresponding pixels. *Decay* is a fraction denoting how quickly the current value should fade over time, and *Intensity* defines how much of the intensity of V_{diff} will be added to the heat map.

The decay and intensity are empirically determined parameters. They were set as follows after testing the visual effect of different values:

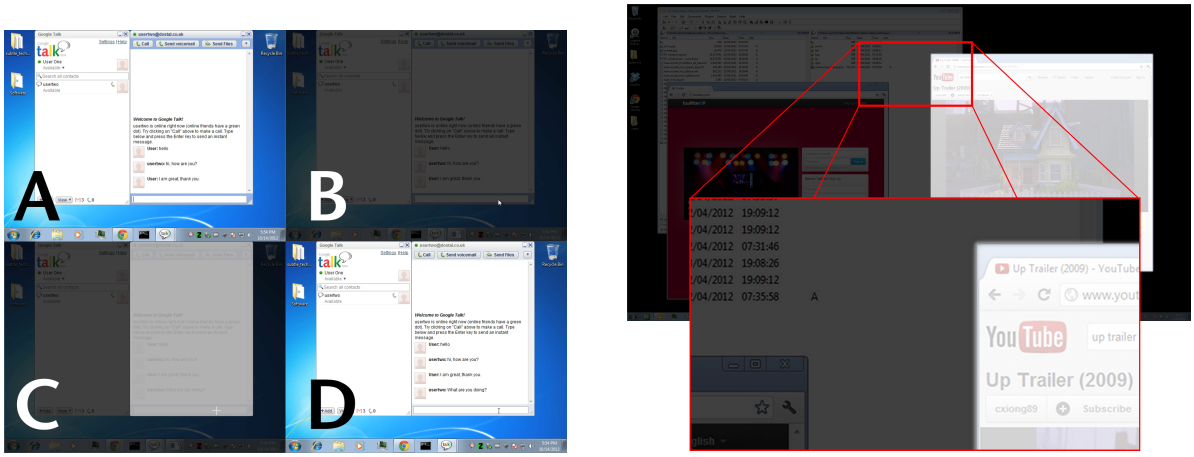
$Decay = 0.01$, $Intensity = 0.5$.

7.2.3 WindowMap

WindowMap is a variation of the PixMap technique. Similarly to PixMap, WindowMap also visualises changes that have occurred on an unattended display. However, instead of visualising changes at the pixel level, WindowMap shows changes at the application-window level. WindowMap works identically to PixMap except that the amount of change on the unattended display is computed for application-window areas on the display instead for individual pixels. Figure 7.4 and Video Figure 1.3 illustrate WindowMap.

The decay and intensity are empirically determined parameters. After testing, the values were set as follows:

$Decay = 0.02$, $Intensity = 0.75$.



(a) (b)

Figure 7.4: An illustration of the WindowMap technique. This technique highlights visual change over time for individual application-windows. *A* shows the last frame before a person shifts visual focus away from this display. *B* shows a frame of the unattended display before any visual change. *C* illustrates how the technique visualises change (a new instant message in the chat window) as it happens by brightening the chat window. *D* shows the current display state after the technique dissolves. Figure 7.4b shows a detailed example of change visualised by brightening the application window, where the change occurred.

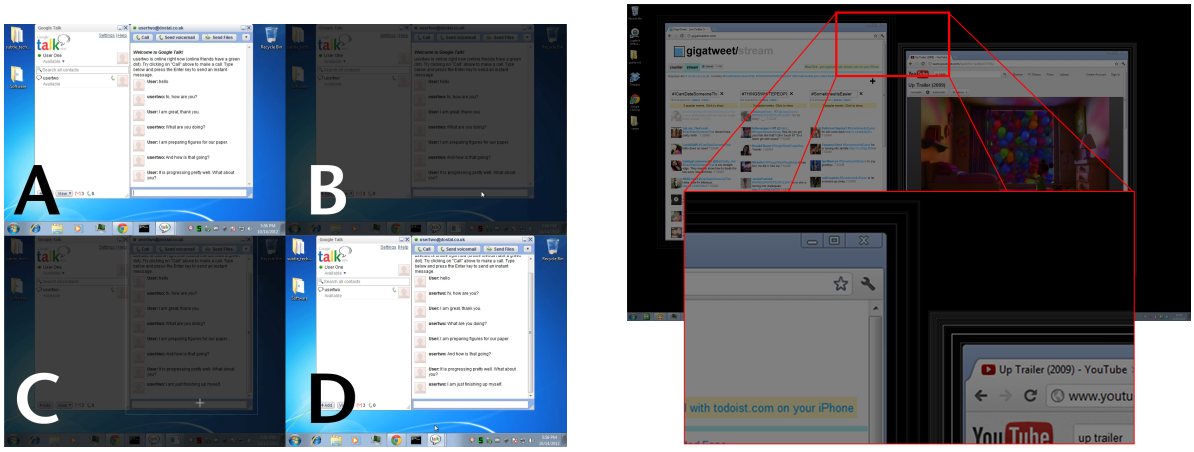
7.2.4 Aura

Aura visualises short-term display changes on an unattended display. Consider a person shifting their visual focus away from display One to display Two. Aura then first darkens display One to reduce distractions. Then Aura continually captures the last twenty frames of the unattended display at approximately one-second intervals. The change for a particular frame is then visualised as a thin rectangle around each window visible on the unattended display One. The brightness of the rectangle for each application window is proportional to how much the window in the frame changed in relation to the previous frame. Since Aura is tracking the changes for the last twenty frames, the visualisation ends up with twenty evenly spaced thin rectangles around each window. The closer a rectangle is to its window, the more recent the visualised change. Figure 7.5 and Video Figure 1.4 illustrate Aura. As the video figure illustrates, a window with many changes (such as a video player) results in cycles of bright rectangles, while a window with few changes (such as an instant messaging window) results in occasional bright rectangles.

7.3 Evaluation

It is difficult to evaluate the effects of subtle visualisation techniques as their behavioural influences require long-term use in realistic contexts. As has been observed in literature, controlled experiments are poor constructs for evaluating such techniques, and a naïve application of a quantitative usability study risks generating misleading results [GB08; Ols07].

Therefore, a formative five-day qualitative case study was conducted instead. A single participant with no prior knowledge of the system or the techniques was recruited. This participant was a PhD student from the same university department and research group. He was 24 years old, familiar with software development and a regular user of multi-display environments. He also wore thin-rimmed



(a) (b)

Figure 7.5: An illustration of the Aura technique. Aura highlights short-term visual changes by projecting a rectangular aura around each window. *A* shows the last frame before a person shifts visual focus away from this display. *B* shows a frame of the unattended display before any visual change. *C* illustrates how the technique visualises the change (a new instant message in the chat window) as it happens as a rectangle around the chat window. *D* shows the current display state after the technique dissolves. Figure 7.5b shows a detailed example of change visualised as rectangles of varying brightness projected around the application window where the change occurred.

glasses. The glasses did not have an effect on the display detector at short to medium distances. This effect only became noticeable at long distances when the eye-pair only occupied a small number of pixels in the camera image. Additionally, prior to the study, the tracking system was extensively tested to ensure it would be able to accurately detect, which display the participant attends to in his multi-display working environment.

The system was evaluated using a workstation configuration consisting of three computers, each with a high-resolution display. Two 2010 iMac computers with 27-inch 2560×1440 pixel displays were used, together with a generic 2010 PC with a 24-inch 1920×1080 display. Each machine was independent, controlling one display. All the machines were connected through a wired network connection and controlled using a single shared keyboard and mouse[1]. It was not possible to drag application windows between displays, but all machines had the same software installed and available. All computers were running the Windows 7 operating system.

A Logitech C910 camera was attached to each display. Each camera provided a constant image stream of 1024×576 pixel images, which was set to be processed by the tracking system at 12 frames per second in order to minimise any performance impact the tracking system might have on the participant's work. Each display was positioned at 76.5 cm from the participant's preferred sitting position. However, the participant was instructed to sit comfortably and there was no restriction on their movement. Figure 7.6 shows the study environment.

In this study, due to the size of the displays, the angular difference between the centres of the 27-inch displays was approximately 45° as measured from the 76.5 cm seating distance of the participant. This angular difference was maintained with the smaller 24-inch display for consistency. This setup represented a prototypical multi-display configuration, as a previous study found that 45° angle between displays is a preferred setup among users [SB05b].

[1]Using Synergy software `http://synergy-project.org/`

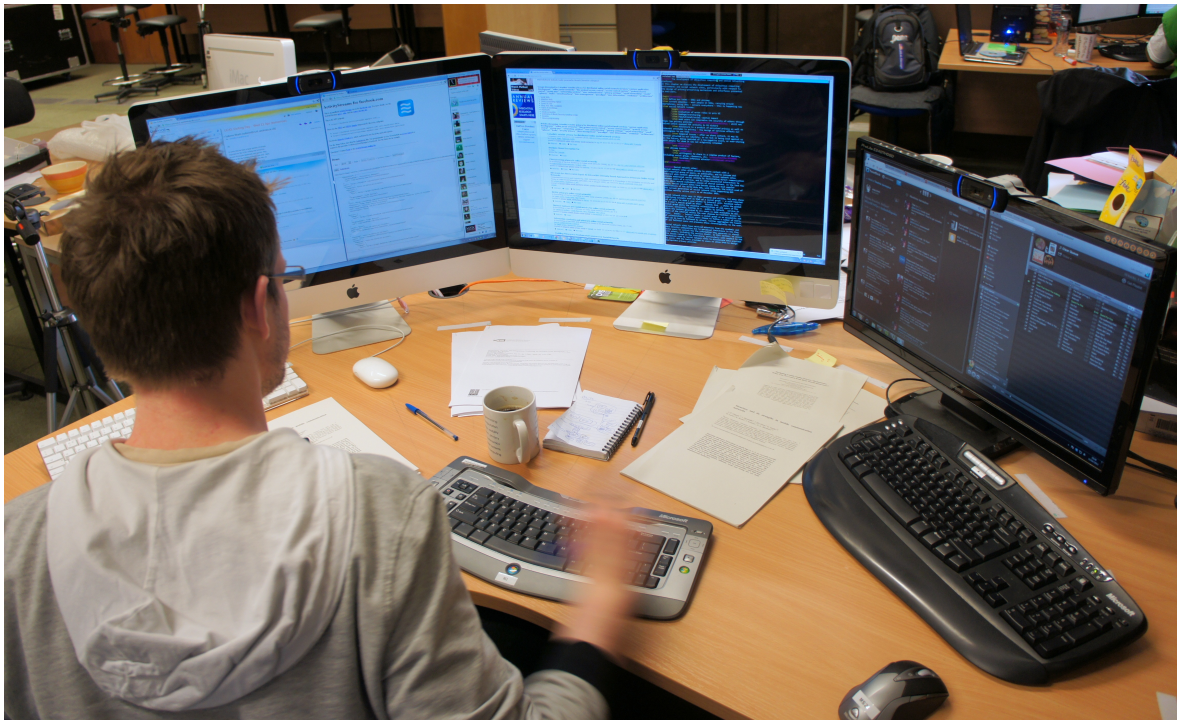


Figure 7.6: A photograph of the display setup used in the study. The two 27-inch displays are bezel to bezel, which translates to approximately 45° angular difference at 76.5 cm distance from the participant.

7.3.1 Method

The participant was asked to use the multi-display environment for five days while performing his usual everyday work tasks. The first day the participant was not exposed to any visualisation technique. This approach was taken in order to obtain a baseline calibration point to compare the subtle visualisation techniques against. For the remaining four days, the four subtle visualisation techniques were deployed in the following order: WindowMap, FreezeFrame, PixMap and Aura. The participant was informed that some variant of an interaction technique that involves multiple displays would be used everyday but the participant was not informed in advance of the particular technique that was used for a specific day. Further, the techniques were not demonstrated to the participant prior to the study.

The system automatically logged the usage of specific displays, the time spent looking at each display, information about attention switches between displays, the start and stop times for the techniques, and the method, by which the technique was terminated (fade or user input). The participant was also instructed to fill out two questionnaires each day. The first questionnaire was administered immediately after the participant finished their work for the day and the second one was administered on the morning of the next day. This meant it was possible to establish both the participant's immediate impression of a technique, as well as his impression of the technique the day after.

To complement the above methodology, the Experience Sampling Method (ESM) was also employed. ESM was perviously proposed as a methodology for evaluating pervasive and ubiquitous computing interfaces [CW03]. ESM enabled collection of data of a higher granularity by presenting the participant with very brief questionnaires at approximately 30 minute intervals throughout their

work day. The questionnaires asked about the participant's current task, his perceived frequency of display switching, and his familiarity with the technique. Finally, the questionnaire included a comment area where the participant could provide additional feedback. Between 9 and 15 ESM samples were collected per day, depending on the participant's physical presence and schedule. Additionally, one of the authors was present in the room for the entire duration of the experiment to administer the ESM questionnaires and to make visual observations.

7.3.2 Results

In total, 39.25 hours of logged usage data (with a one second sampling rate), 9 questionnaires, and 61 ESM samples were collected from the participant.

Display Utilisation The participant spent a total of 31.66% of his time not attending to any of the displays due to various reasons such as meetings or paper-based work. Figure 7.7 shows the participant's display utilisation during the study as recorded in the system logs, adjusted to active-use time. Figure 7.7 makes it clear that the participant primarily attended to the centre display. This data suggests that the additional screens are mostly used for secondary or peripheral tasks. This is in line with previous research on the utilisation of multiple monitors [Gru01]. In addition, based on the collected ESM samples and visual observations, peripheral display tasks appear to be mostly related to social networking (Facebook, Twitter), instant messaging (MSN) or media streaming (Spotify). The peripheral displays were also commonly used as a point of reference for information such as documentation, server logs or an application programming interface (API). In the collected ESM samples, the participant also referred to the displays on either side, as well as activities performed on them, as peripheral.

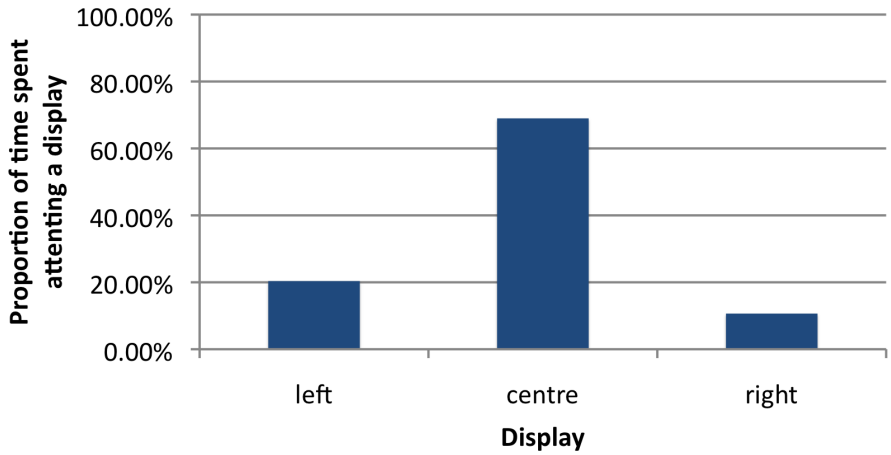


Figure 7.7: The distribution of display usage adjusted to active-use time.

Display Switching Figure 7.8 plots the mean number of switches per hour of use for each technique. All four subtle visualisation techniques reduced the frequency of display switching by approximately a third or more compared to the baseline (which used no technique). FreezeFrame reduced the frequency of display switching by the smallest amount, only 32.5%. This may be due to FreezeFrame's inability to dynamically visualise display change. This means that the only mechanism for a person to check if there has been a change on the display is to switch visual focus to that display. Aura reduced the rate of display switching by 37.9%. This may be due to the rectangular "aura" around

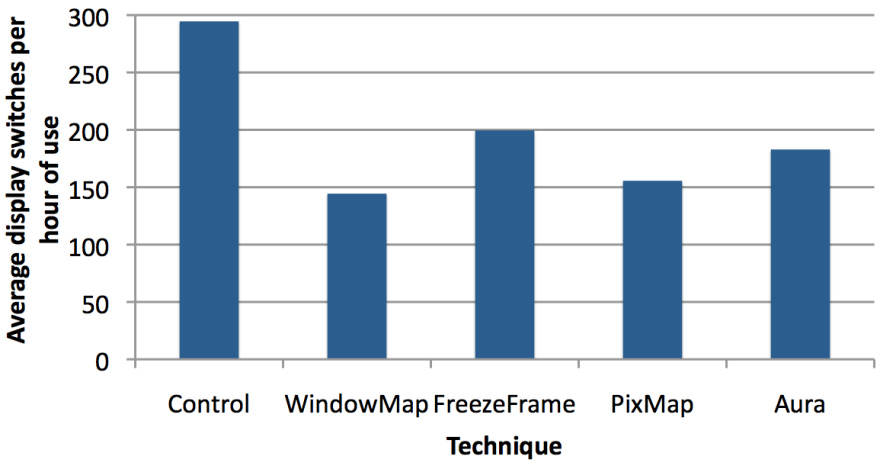


Figure 7.8: The average number of display switches per hour of use for each of the techniques.

the windows being too dim for the participant's peripheral vision to efficiently perceive it. Both PixMap and WindowMap reduced the rate of switching the most (47.2% and 51%, respectively). The similarity between the results of these techniques was expected due to their similar nature. A possible explanation may be that PixMap and WindowMap visualised change in such a way that the participant could use his peripheral vision to only react to changes that warranted his attention.

The analysis of the subjective data suggests the participant had difficulty estimating his frequency of display switching. According to the end-of-day questionnaires, the participant believed he was intensely switching displays during the day. His median rating on a seven-point Likert-type scale ("How often did you switch displays today?" 1 = Not at all, 7 = Very often) was 7, and his minimum rating was 5. However, the ESM samples overall median rating for the study was 4 on the same seven point scale. While this still indicates that the participant felt he switched displays at least moderately frequently, the ESM rating is markedly lower. The participant perceived the display switches that happened during the work day as being more frequent at the end of the work day compared to the ESM samples taken throughout the day. Further, while the participant reported a high rate of display switching at the end of each day, based on the ESM samples he was subjectively display switching less often when the subtle visualisation techniques were deployed.

The high number of display switches shown in Figure 7.8 might be related to distractions caused by applications on the peripheral displays. The participant repeatedly noted that he was often distracted by pop-up notifications and other animated content on the first day of the study (the day none of the subtle visualisation techniques were used). On the other hand, the participant did praise all the techniques for their distraction-reducing qualities, be it through desaturation, decrease in contrast/dimming, or by only highlighting areas of the screen where a change was happening. This subjective data corroborates the measured frequency of display switching collected from the system logs (Figure 7.8).

Tasks and Display Switching Information about the primary task the participant was performing at the time of the sample was collected as part of the ESM sampling. This provided coarse-grained information about the temporal distribution of the tasks during the study. Each task (or application use) specified by the participant was classified into a broader task category (see Table 7.1 for a list of task categories). Most of the task categories are self-explanatory. *Academic Reading* encompasses reading academic papers in PDF format and searching for papers online. *Internet Browsing* includes all other activities that use the browser, most often reading blogs and using social media.

Since the ESM samples were administered at approximately 30 minute intervals, the system logs with display switching data were divided into 15 minute segments. The timestamp of each ESM sample was used as a centre of the period during which the specified task was performed. This enabled estimation of the time spent on each of the specified primary tasks. This temporal information was linked to the number of display switches. Aside from the coarse estimation, a further limitation of this data was that there was little information about the secondary tasks performed. From the ESM samples and the observations of the study administrator, the secondary tasks most often seemed to include a combination of internet browsing and social media use, instant messaging, media playback and server management through a terminal application.

Due to the unrestricted use of the computers, not all tasks were performed every day. Therefore, the data for all techniques was aggregated together. While this did not allow for fine grained comparisons between techniques, it did allow for exploration of general trends towards distraction reduction for specific tasks (as all the subtle techniques should reduce distractions to some extent) and for comparison of different tasks with regards to the amount of display switching.

Task	Switches		Time [mins]	
	total	per min	active	total
Academic Reading	1035	2.43	425.42	540
Technical Reading	94	2.35	40.05	45
Graphical Editing	1207	3.33	362.72	450
Text Editing	583	2.45	237.52	315
Internet Browsing	200	3.07	65.08	105
Instant Messaging	308	5.13	60.08	105
Email	52	4.08	12.73	30
Other	133	3.19	41.72	45

Table 7.1: Display switches for specific tasks while the subtle visualisation techniques were running.

Table 7.1 shows the frequency of display switching for all the primary tasks performed during the four days the subtle techniques were used (as that is the larger, more diverse dataset). Apart from the total number of switches for a particular task, the table also shows the mean number of switches per minute of active computer use. In addition, the table shows the estimated total duration of each task over the four days, as well as the active computer usage time. The tasks with the highest mean number of switches per minute are the tasks which either tend to lend themselves to multi-display use (e.g. *Graphical Editing* with source information on one display and generated graphics on another) or tend to be of low importance, or glance-able (e.g. *Instant Messaging* and *Internet Browsing*). The high mean number of switches for email may be due to the short overall amount of time spent on the task and thus the result may not be very representative.

Task	Switches [per min]		Change
	Control	Techniques	
Graphical Editing	4.30	3.33	-22.62%
Text Editing	5.65	2.45	-56.56%
Instant Messaging	5.31	5.13	-3.40%

Table 7.2: The effect of subtle visualisation techniques on specific tasks.

Table 7.2 shows a comparison of three tasks that were performed both during the control day and the four days with subtle techniques. The table reveals the mean number of switches per minute for the two datasets. The subtle techniques decreased the number of display switches in all cases.

In the case of *Instant Messaging* the change was small (3.4% decrease), which is likely due to the glance-able nature of instant messaging. However, for both of the editing tasks, the decrease is much more pronounced. The decrease for *Graphical Editing* is lower (22.62%) than for *Text Editing* (56.56%). This is probably because *Graphical Editing* is more expensive in terms of display space and thus is more likely to be used as a multi-display task, therefore requiring more frequent switches between displays. In either case, it can be concluded that the subtle techniques successfully reduced the frequency of display switching, especially for editing tasks.

Qualitative Analysis In addition to investigating display utilisation and display switching, the participant was interviewed about the perceived usefulness of the techniques. The first technique the participant used was WindowMap. Observations by the study administrator indicated that the technique helped reduce distractions and allowed the participant to better focus on his primary task. For example, the participant stated that *"Dimming peripheral screens makes it easier to focus on [the main screen]."*

However, the timing design of WindowMap confused the participant. WindowMap was set up so the visualised changes would decay after approximately 20 to 30 seconds if no other changes occurred. This seems to have led to a situation, where the participant was noticing a change but was unsure about the timing of when the change occurred and whether or not it was continuous. However, a possible reduction in cognitive load was noticed when the participant processed events from the peripheral displays. For example, the participant stated that *"while waiting for a window to finish processing something, I was able to note the state change in my peripheral vision, which was more useful than randomly glancing over to check if it was done."* The participant ranked this subtle visualisation technique third in terms of perceived usefulness.

The second technique the participant used was FreezeFrame. FreezeFrame is a technique for re-establishing prior context before and after a display switch. FreezeFrame requires full visual focus and does not rely on people being able to notice changes in their peripheral vision. The qualitative data indicated that this behavioural design of FreezeFrame helped the participant monitor longer-term changes more effectively: *"I find it very useful for more passive applications such as twitter. When glancing over, I have an immediate sense of what tweets are new, without being constantly distracted."* However, the fact that it required the participant's full visual focus meant that checking peripheral yet time-sensitive applications, such as an instant messaging conversations, created more of a distraction: *"I've found it useful to be able to glance at contact lists to see who's come online recently, because the change in the list is so easy to perceive, without distracting me while working."* Finally, while the participant felt the time-to-fade was too long, it was still considered the second most useful technique.

The third technique the participant used was PixMap. As previously noted, PixMap is similar to WindowMap in that it visualises changes over time. However, while WindowMap visualises display changes over time on the application-window level, PixMap visualises display changes over time on the pixel level. The participant perceived PixMap as more useful than WindowMap because it provided him with more fine-grained information, while also being more subtle. The participant reported that PixMap was particularly useful for applications that only resulted in smaller display changes, such as list updates: *"The 'trail' effect is particularly useful for gauging how lists have changed, such as on twitter - giving a glanceable way of telling how many new tweets there are."* The participant ranked PixMap as the most useful of all the subtle visualisation techniques.

The last subtle visualisation technique the participant tested was Aura. The participant felt it was useful for reducing distractions: *"Not distracting - keeps me focused!"*. Aura was perceived as very subtle, sometimes too subtle. In part this was because the visibility of the visual "pulses" that visualised the amount of change in the window at each of the past time steps depended on the background colour of the application. Since high activity was visualised as a white colour, this meant that when the background was white, it became difficult to perceive the change: *"I'm aware of change*

in some apps more than others. For example, one screen has two apps with dark colour schemes so the 'pulse' effect is more pronounced, whereas with apps with white backgrounds, I often miss any subtle change." The participant ranked Aura as the least useful subtle visualisation technique.

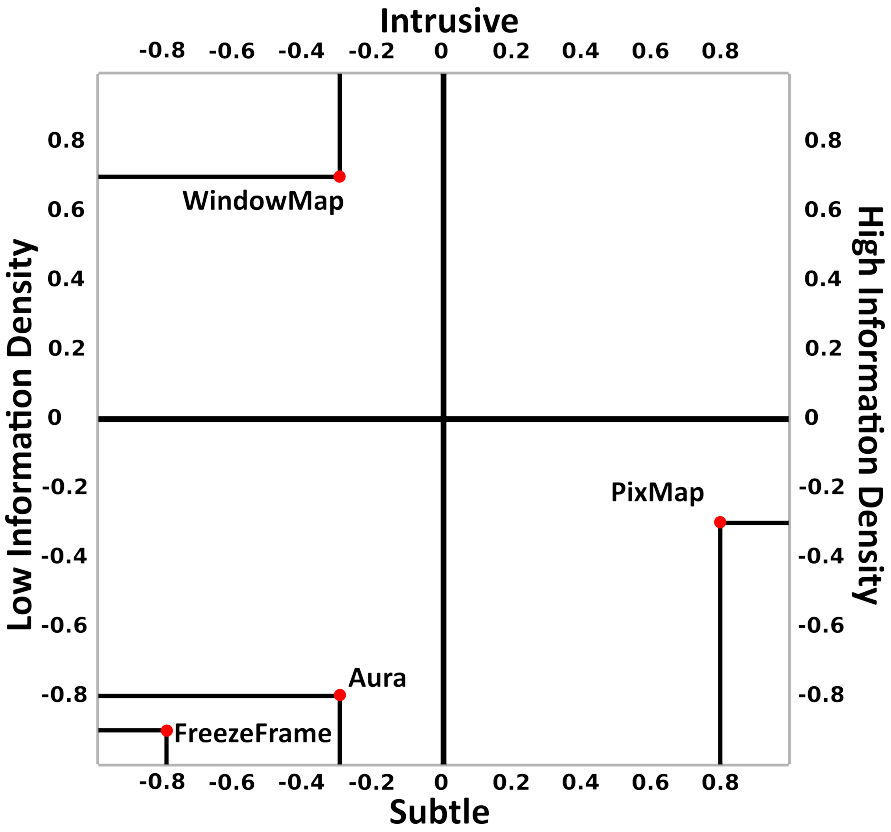


Figure 7.9: Mapping of all the techniques based on their perceived subtlety/intrusiveness (y-axis) and information density (x-axis).

Subtlety and Information Density Figure 7.9 shows the participant's subjective mapping of the techniques in terms of their subtlety or intrusiveness and information density. Information density is a measure of the compactness of a particular technique in terms of the amount of useful information the technique displays. Notably, the ordering on the subtlety axis closely corresponds to the ranking of perceived subtlety reported by the participant in the daily questionnaires.

There was a separate question in the questionnaires about perceived contribution to increased productivity. Its ranking was exactly the same as for perceived usefulness. Since periphery-enhancing information is seen as one of the indicators of calm technologies [WB95], the perceived usefulness of a technique was expected to correspond to its perceived information density. However, this was not the case as the ranking of perceived usefulness was No Technique $<$ Aura $<$ WindowMap $<$ FreezeFrame $<$ PixMap, while for information density it was No Technique $<$ FreezeFrame $<$ Aura $=$ WindowMap $<$ PixMap. This means that while FreezeFrame was ranked low on information density, it was perceived as very useful.

7.3.3 Limitations and Future Work

One limitation with this work is the way the system detects people's visual focus on a particular display. The tracking system depends on inter-device angular difference. Since the system relies on a minimum angular difference between displays for accurate visual focus detection, people's multi-display environments may need to be modified to accommodate this. This can be done by either changing the size of displays, or their distance from the person.

Another limitation concerns the study used to evaluate the subtle visualisation techniques. In order to fully understand the techniques, an entire set of evaluations would be needed. To frame the discussion, we will use the work of Matthews et al. [MRC07], who introduced an approach to evaluating peripheral displays that is based on Activity Theory. As part of their approach, they introduced five metrics that designers and evaluators could focus on - appeal (usefulness, aesthetics), learnability, awareness, effects of breakdowns and distraction. With most of the metrics, Matthews et al. note that the effects of a particular design is heavily influenced by the activity supported by the design. Because the visualisation techniques were designed as generally applicable they have can potentially support a very broad scope of uses, this makes designing evaluations complex. Therefore, the study presented in this chapter should be seen as a first step towards more detailed evaluations of the techniques.

The main advantage of the chosen evaluation format is that it provides in-depth information about how the techniques were used by a single person. The results include information about the activities that the person performed as well as some insight about the relative effects on display switching. The study environment was also designed to match the environment the participant was used to, giving the study ecological validity. However, since the study only included a single participant and did not limit the activities the participant performed, we cannot know how well the results of the study translate to different people and different combinations of activities performed.

Therefore, the study presented in this chapter should be taken as a starting point informing a series of more controlled, in-depth evaluations. We suggest several follow on evaluations and variables that could be included in the design.

Since the use was not constrained in the initial study, a further study of the effect of the techniques on specific tasks/activities would be valuable. A controlled lab study with a number of participants would be well suited for this. The results of the initial study could be used to select the task/activities to evaluate, for example by selecting tasks with the greatest difference of display switches (instant messaging and technical reading). It would then be possible to create a scenario, in which the specific effects of the techniques could be measured.

For example, to evaluate the effects of the techniques on monitoring secondary activities, a scenario could involve the participant performing a primary activity (e.g. technical reading) performed on the primary display, with a secondary activity (e.g. instant messaging) on the peripheral displays. The level of awareness of changes in the periphery using the different techniques could be measured with knowledge questions after use or by observation during use.

Another controlled study may be used to evaluate the amount of distraction caused by the techniques. In this case, the primary activity should be chosen to be highly demanding on the participant's focus, e.g. playing a game that requires the participant to race a car on the primary display. The secondary displays could then be set up with information that could potentially be passively monitored, e.g. updated on a twitter feed. The influence of the peripheral activity on the primary task in terms of completion time could then be measured.

To summarise, the study presented in this chapter has a relatively high validity in terms of simulating real-world use, but the generalisability of the results is limited due to the unconstrained nature of the activities and only using a single participant. A combination of more focussed controlled laboratory studies and an increased number of participants would help confirm the validity of the results and increase their general applicability.

	Distance					
Name	AA	AO	AE	OO	OE	EE
DiffDisplays		P				

Table 7.3: Relationship - Distance

Relationship legend: A - Actor, O - Object, E - Environment; e.g. AO — Actor-Object
Value legend: C - Continuous adaptation; D - Discrete, zone or threshold based adaptation; P - Binary adaptation based on presence/absence

	Orientation								
Name	AA	AO	AE	OO	OA	OE	EE	EA	EO
DiffDisplays		D							

Table 7.4: Relationships - Orientation

Relationship legend: A - Actor, O - Object, E - Environment; e.g. AO — Actor-Object
Value legend: C - Continuous adaptation; D - Discrete, zone or threshold based adaptation.

	Range			
	intimate	personal	social	public
Name	(<0.6m)	(<1.5m)	(<5m)	(>5m)
DiffDisplays	x	x		

Table 7.5: Interaction - Range

Value legend: x - System was described or demonstrated in use within this distance range

	Cardinality			
Name	Actor	Object		Environment
DiffDisplays	1	1	1*	

Table 7.6: Interaction - Cardinality

	Mode							
Name	Actor				Object			Environment
DiffDisplays	Sp	Vis	Int	Sym	Sp	Vis	Int	Sym	

Table 7.7: Interaction - Mode

Value legend: Sp - Spatial, Vis - Visual, Int - Intent, Sym - Symbolic, Ac - Acoustic

7.4 DiffDisplays and the IRE Model

DiffDisplays is a prototype system used to evaluate visualisation techniques that show changes during periods of inattention. The main purpose of the techniques is to visually convey change that occurred while a person's attention was directed away from a display. Technically, the system forms a single or multi-display environment with each display being augmented by a web camera. The camera performs computer vision tracking of the person's eyes, which helps estimate whether they are looking in the direction of a specific display or not. When the person is looking at a display, the system lies dormant. After their attention shifts elsewhere, the system activates a visualisation technique, which tracks any visual changes on the unattended display. When the person's attention

	Intentionality			
Name	Actor		Object	Environment
DiffDisplays	E	I	E	

Table 7.8: Interaction - Intentionality

Value legend: E - Explicit, E+ - Explicit (with possible misclassifications), I - Implicit, I+ - Implicit (with possible misclassifications), A - Ambient

	Intensity			
Name	Actor	Object		Environment
DiffDisplays	N	U	S N	

Table 7.9: Interaction - Intensity

Value legend: U - Unnoticeable, S - Subtle, N - Neutral, I - Intrusive, D - Disruptive

returns to the display, the visualisation slowly fades into the live state of the desktop, giving the person a chance to quickly establish regions where change has occurred.

7.4.1 Entities and Relationships

In *DiffDisplays*, there are two types of entities - Actor and Object. The displays on the work desk are Objects being interacted with and the person interacting with them is the Actor. Spatial interactions with *DiffDisplays* are limited but essential to the system. Orientation of the Actor towards the Objects is used are a proxy for visual focus, which means that the Actor-Object orientation relationship is utilised. Due to the way the displays are arranged, only one of the displays can be the centre of attention at a time, resulting the the relationship being classified as a discrete one. Additionally, the presence or absence of the Actor in the interactive space is also used to trigger the visualisation techniques and and maximum detection distance is used as a threshold, so the Actor-Object distance relationship is used, in its presence/absence form. Tables 7.3 and 7.4 summarise this information.

7.4.2 Interaction

Interactions in *DiffDisplays* only occur within the intimate and personal ranges. This is partly due to the sensor setup with the tracking system setup to track distances only up to 2 metres to minimise the use of compute resources. Additionally, it is due to the use case where interactions are expected to occur while a person is sitting on a chair at their desk, which naturally limits the interaction range. Table 7.5 shows the range values.

The scenario of use as well as the tracking system assume a single person interacting with the system, which means that only single Actor interactions take place within the system. The display setup is flexible where a single display can be used if needed, even though the system has been primarily setup as a three display setup. Even though the keyboard and mouse input are shared between all the displays, each of the display is running independently of the others in terms of tracking and decisions about triggering the interaction techniques. This means that the most common cardinality value for the system is `1*` as the three displays are used in parallel. Table 7.6 summarises the values.

The values for modes are shown in Table 7.7. The use of mouse pointing by the systems is classified under the intent mode. The use of visual information on the displays, together with textual

information means that visual and symbolic modes are also used. The use of orientation changes to trigger the interaction techniques means the spatial mode is used as well.

Table 7.8 contains the intentionality values for *DiffDisplays*. Starting with the displays (Objects), all the actions by the displays are *explicit* within the IRE model. The Actor's actions while interacting with the displays through the keyboard and mouse are classified as *explicit* as the input methods are generally robust to accidental input. The changes in the Actor's orientation towards are interpreted by the system as a proxy for visual attention. This is performed no matter what the primary purpose of the orientation change is, which means that the action is an *implicit* one within the IRE model. Because the Actor needs to change their orientation by a large amount in order to reach the threshold for activating one of the visualisations, the sensing input is sufficiently robust to not be prone to misclassifications of the Actor actions.

Lastly, the intensity values are summarised in Table 7.9. For the Actor, actions are classified as *neutral* as they clearly impact the system, but the magnitude of the impact is exactly as significant as it was intended. For actions by the displays (Objects), the occupy the entire spectrum between *unnoticeable* to *neutral*. Since the visualisation techniques are not triggered until after the person looks away from a particular display, the visualisations may not be visible to the Actor at all for periods of time. However, depending on where the Actor is looking parts of the visualisations may be visible in their peripheral vision and (by design) in a relative low-impact manner (so classified as *subtle*). When the visual focus of the Actor returns to a display, the visualisation is stopped and removed from the display. This clearly impacts the Actor but the process of removing the visualisation has been designed to not be overly intrusive, so this last stage of the interaction with the visualisation is classified as *neutral*.

7.5 Conclusions

This chapter introduced *DiffDisplays*, a system built to explore subtle gaze-dependent techniques for visualising display changes in multi-display environments. Additionally, four subtle techniques for visualising changes in unattended displays were presented: FreezeFrame, PixMap, WindowMap and Aura.

The efficacy of the visualisation techniques was studied in a five-day evaluation with a working professional. This participant used the system eight hours per day for five consecutive days. The results of the study showed that the techniques were successful in highlighting visual change, and in reducing distractions and the frequency of display switches for the participant. Further, three of the four techniques did so in a non-intrusive and subtle manner.

This chapter also revealed challenges when evaluating subtle intelligent interaction techniques that require deployment in actual working contexts in order to generate meaningful data. The evaluation in this chapter used a combination of questionnaires, ESM sampling, observations and automatic logging. This methodology proved successful for the evaluation presented here, but it is also prohibitively expensive for a large-scale study.

The results of the evaluation suggest that subtle visualisation techniques can positively change the display attendance behaviour of a working professional who used these techniques as part of his regular work activities. However, many opportunities still exist for taking the technology further, both in terms of further refining the techniques, and in terms of identifying cost-effective means of evaluating the efficacy of such techniques. The subtle display of visual change information from unattended displays can alter our interactions and expectations about interfaces. An extension of the work presented in this chapter, which utilises the same underlying system but focuses on increasing peripheral awareness has been developed by Garrido et al. [GP+14b; GP+14a].

To conclude, *DiffDisplays* demonstrates the last of the facets of interactions with displays. The *MultiView* prototypes in Chapter 5 showed it was possible to create visual spatial interactions with

displays without the need for tracking by exploiting the properties of the display to generate multiple simultaneous spatially separated views. *SpiderEyes* in Chapter 6 explored the breath of possibilities for dynamic visual content manipulation when tracking is available. *DiffDisplays* and this Chapter demonstrate that spatial interactions (in this case orientation as a proxy for visual focus) can play an important role for interactions that occur when a person is *not* looking at displays.

Chapter 8

Summary and Conclusions

With the increasing number and diversity of devices we interact with, the character of our interaction with them is changing. Spatial interactions are becoming more commonplace and the range of sizes, types and positioning of displays we interact with is ever changing. This thesis focuses on the intersection of the two topics, primarily exploring aspects relating to visual spatially-aware interfaces and systems.

8.1 Thesis Summary

This thesis first introduced the Interaction Entity Relationship (IRE) model, which is an interaction model that primarily focuses on spatial relationships between interactive entities. Additionally, the model covers the classification of three types of interactive entities as well as a set of characteristics related to interaction.

Next, the IRE model was used as a lens for analysis of existing systems utilising spatial interactions. The analysis revealed a number of gaps and opportunities, some of which are detailed further in this chapter in Section 8.4. The analysis also provided a mechanism to position the three case studies presented later in the thesis within the broader context of the research field.

Chapter 4 presented the technological platform, which was used to implement the prototype systems in this thesis. The main focus of the chapter was on computer vision based algorithms, which allow the use of common hardware such as RGB cameras to provide spatial information about a tracked person, with specific emphasis on distance and coarse grained orientation. This chapter also included a number of evaluations demonstrating different functional and performance characteristics of the tracking algorithms.

The three chapters that followed each contained a case study of one of three aspects of spatial interactions with displays. The first case study, *MultiView* used the limited viewing angles and associated colour and contrast compression of a common type of LCD display to create a multi-view display capable of simultaneously displaying two distinct views of on-display content. These multiple views were used to create spatial interactions that did not require explicit spatial tracking. A validation study of two prototype systems was also included in this chapter.

The second case study, *SpiderEyes*, concentrated on dynamic manipulation of on-display content on a single large display. As part of the case study the possible combinations of content manipulation techniques were systematically explored. This exploration resulted in the presentation of the Content Manipulation Matrix. Several of the gaps were explored through example scenarios, which were later implemented using a prototyping tool. In addition to content manipulation techniques, the prototyping tool also allows its user to explore other spatial interactions.

The third case study, *DiffDisplays*, explores visual spatial interactions that occur during periods of inattention. *DiffDisplays* uses coarse grained orientation tracking to trigger visualisation techniques for tracking visual changes on displays while a person is looking away from the display. The techniques were evaluated in a five-day qualitative study.

8.2 Research Questions

The introduction of this thesis stated the central hypothesis, which is that *studying how spatial relationships can be leveraged in indoor environments for interactive purposes can enable development of novel interactions*.

This hypothesis has been investigated by answering the following four research questions:

Question 1: How can the use of spatial relationships in existing interactive systems be analysed? This question is answered through the creation of the IRE model in Chapter 2. The model places a particular focus on the types of interactive entities and the spatial relationships between them.

Question 2: How are spatial relationships currently used by existing indoor systems? This question is addressed by analysing a number of existing research systems using the IRE model in Chapter 3.

Question 3: How can the prototyping of spatially aware indoor visual interfaces be supported? This question is answered in two ways. Firstly, Chapter 4 summarises the design and implementation of several computer vision-based tracking algorithms. Secondly, the *SpiderEyes* toolkit presented in Chapter 6 can facilitate faster prototyping of spatially-aware visual interfaces.

Question 4: Can the prototyping support tools created in this thesis be used to create novel interactive systems? This question is answered through the creation of four different prototype systems, each of which has a different focus. The two *MultiView* prototypes introduced in Chapter 5 demonstrate that it is possible to design systems that enable spatial interactions that do not always require active tracking. The *SpiderEyes* system in Chapter 6 concentrates on the use of on-display content manipulation. Lastly, the *DiffDisplays* prototype in Chapter 7 shows how indirect use of spatial relationships can be leveraged to create novel interactions.

8.3 Contributions

Below is a summary of contributions in order of appearance (with several minor contributions omitted):

- A descriptive and comparative interaction model primarily based around spatial relationships between interactive entities. Using the model to analyse existing systems revealed a number of trends and gaps in existing research.
- A number of tracking algorithms that allow for distance and coarse orientation tracking of people for interactive purposes, using low-cost, high-availability hardware and software.
- Three case studies, exploiting different visual properties of displays and people for interactive purposes in order to deepen our understanding of their interactive expressiveness. Each of the case studies was based on a distinct set of constraints.

MultiView demonstrated how visual properties of displays can be leveraged to generate multiple simultaneous views, which can be used to create interactive spatial interfaces that do not require dynamic manipulation of visual content to enable spatial interactions.

SpiderEyes revealed and subsequently explored the design space for dynamic manipulation of visual content on displays for interactive purposes. The prototyping tool presented with SpiderEyes and the Content Manipulation Matrix form additional contributions.

DiffDisplays showed that interactions during times where visual content on displays is not visible also provide a valuable avenue for research. The subtle techniques for tracking visual change on unattended displays are also a contribution.

8.4 Implications and Future Work

The work in this thesis has lead to a number of observations about existing research systems as well as revealing potential research opportunities. This section discusses possible implications for researchers and system designers and outlines some of the potentially more significant avenues for future research.

8.4.1 Implications for Researchers

Research System Descriptions

Reproducibility of research can be a concern with publications. While conducting the analysis in Chapter 3, it became clear that with interactive systems, there are potential concerns beyond replication. Sensing methods and other technologies, on which interactive systems are built, still change rapidly, which leads to difficulties in replicating interactive systems. However, since the focus of many publications is on interaction techniques rather than implementations, the technical details are arguably not as important as detailed descriptions of the interaction techniques. If an interaction technique is described in sufficient detail, it would likely be possible to replicate it using some form of currently available technology, even if the technology used to originally implement the technique is no longer available.

The IRE model is a relatively high-level model, concentrating on the presence or absence of spatial relationships, or high-level interaction characteristics, rather than more granular details of interaction techniques or atomic actions. However, the analysis in Chapter 3 revealed the difficulty in conducting even a high-level analysis of existing systems due to the lack of specificity and detail in the descriptions of the system and its interaction techniques.

The IRE model offers a lens through which interactive systems can be described. Using the model or a similar conceptual framework to position and describe systems can potentially increase the quality of system descriptions, so researchers and system designers are encouraged to take advantage of it in their publications. Moreover, the absence of an established and reliable standard for descriptions of interactive systems presents a potential research opportunity as the standardisation of required detail within descriptions of interactive systems would be a valuable contribution.

Physical and Visual Constraints

In addition to considering the interactional environment, researchers and designers would likely benefit from taking into account the physical and visual constraints of all the interactive entities. One of the motivating factors could be to avoid introducing potential sources of noise into experimental results due to physiological or technological constraints. Moreover, actively designing with the constraints of all interactive entities in mind could lead to more usable systems. Additionally, as demonstrated in Chapter 5, it could potentially open up new possibilities for interactions. The

Appendix of this thesis contains pointers to relevant literature as well as a number of summaries, data points and heuristics relating to human vision, displays and how they may interact within spatially-aware systems.

8.4.2 Future Work

The work in this thesis is by necessity limited in its scope due to available time, methods and other constraints. While the research presented in this document stands on its own merit, the work can be extended in various ways. Additionally, the results of the explorations and analyses in the last six chapters revealed a number of additional avenues for future research. This section summarises the most significant opportunities.

Proxemic Interactions

During the analysis in Chapter 3, it was revealed that a number of existing spatially-aware systems were motivated by the notion of proxemics as introduced by Hall. Proxemics is concerned with how people use space in interpersonal communication [Hal66]. However, the results of the analysis showed that most of the systems concentrated on Actor-Object relationships rather than interactions between Actors.

While the use of Actor-Actor spatial relationships was one of the more frequently explored relationships, the use of this relationship was mostly to make assumptions about whether two or more Actors should be considered as a single interactive group. The nuances and intricacies of interpersonal proxemic interactions within the context of spatially-aware interactive systems have not been explored in depth, presenting an opportunity for future research.

Interactions Beyond Actors and Objects

As described previously, the analysis in Chapter 3 revealed a large number of gaps in the use of spatial relationships by existing systems. While most of the relationship combinations were used by at least one existing system, there appears to be a large amount of relatively under-explored spatial relationships.

Starting with relationship types, orientation was the least used spatial relationship overall. Moreover, systems that make use of orientation mostly use it either as a proxy for visual attention or to optimise the view of on-display content for a person's viewpoint. Since there are so many gaps, systematic exploration of potential uses of entity orientation may reveal additional opportunities for the use of orientation in interactive scenarios.

In terms of entity types, the most under-used entities were Environment entities. This is somewhat linked with the observation that researchers and system designers do not seem to explicitly consider the interaction environment in their designs. However, the research opportunity here lies in the observation that the use of Environments as interactive entities has not been broadly researched, whether as a systematic mapping out of the design space, or in terms of specific interaction techniques.

Visual Content Manipulations

The *SpiderEyes* case study investigated the possibilities for dynamic manipulation of visual content, leading to the introduction of the Content Manipulation Matrix. However, while interactive potential of some of the discovered gaps has been validated and qualitatively explored through the example scenarios as well as through their implementation, a more quantitative evaluation would likely strengthen the initial results.

Inattention and Unintentional Interactions

This research opportunity is motivated by results from two of the chapters, namely the analysis results from Chapter 3 and the third case study exploring visualisations on unattended displays in Chapter 7. The case study showed that considering inverse interactions, such as focusing on the absence of visual attention rather than the presence of it, can lead to novel research opportunities.

In addition, the results of analysis of interaction intentionality in Chapter 3 revealed a research gap with unintentional interactions. Investigating, for example, how unintentional actions of persons could be used to provide implicit information about the person may lead to valuable insights.

8.5 Concluding Remarks

The complexity and variety of our interactions with displays is increasing. It is not uncommon for a person to have at least three or four devices with displays with a wide range of sizes (e.g. a mobile phone, a tablet, a desktop computer and a TV). The typical interactions with these devices occur at different distances. Additionally, as our interactive space become less constrained, the importance of the spatial aspects of our interaction increases.

The work in this thesis demonstrates the potential of visual interactive systems that use spatial relationships. It builds on previous research to deepen our understanding of how different aspects and constraints of the world around us, including ourselves, can be leveraged to create richer, more flexible interactions. Moreover, it adds to this knowledge by providing functional examples and approaches to building systems that make use of the findings. We could be entering an era of interactions with adaptive systems, that can have the ability to optimise our experience based on our personal technological and spatial constraints, whether they are temporary or permanent. This thesis provides a stepping stone for making this vision a reality.

Chapter 9

Appendix - Visual Considerations for Spatially-Aware Systems

This appendix introduces a number of considerations for designer of future systems, whether they be primarily multi-display environments or spatially-aware systems.

There is a wealth of research of different aspects of human vision, and covering it in any significant depth is beyond the scope of this thesis. However, the purpose of this appendix is to highlight the importance of considering the specific characteristics and limitations of the people using the systems we build as well as the objects, such as displays, that form the interactive systems. A focused but relatively high-level overview of existing literature is sufficient for this purpose.

This appendix contains potentially unfamiliar terminology. The list below provides brief glossary of the most important terms used in this appendix:

temporal in the direction of the temple (the side of the head); towards the temple

nasal in the direction of the node; towards the nose

monocular visible by a single eye

binocular visible by both eyes

ambinocular visible by at least one eye

FOV field of view

DOF depth of field

9.1 Human Vision

This section provides an overview and summary of the most relevant aspects of the human vision system. Specifically, the focus is on the field of vision, the movements of the eye and head, visual acuity and depth of field. A large body of reserach on human vision is available, from the optical characteristics of the eye to the way visual stimuli are processed by the brain. The summary presented here captures a small subset of existing literature, mainly concentrating on visual acuity and related characteristics. A similar characterisation of colour and shape perception could prove beneficial, but this is out of scope of this thesis. The purpose of this appendix is to demonstrate the importance of considering human and display characteristics and limitations, which is sufficiently substantiated by focusing on mainly visual acuity and related characteristics.

9.1.1 Field of Vision

The field of view is the area that is seen by our eyes at a particular moment. Monocular field of view is a field of view seen by a single eye (or at least a single eye in some circumstances) and a binocular field of view is the area seen by both eyes. Ambinocular field of view is the field of view, which is seen by at least one of the eyes. The figures and values for the field of view limits are according to the Bioastronautics Data Book [PW73].

For a single eye, the field of vision limits are approximately 95° temporally, 60° nasally, 46° up, and 67° down. Figure 9.1 shows the shape of the field of view for a single eye. The nasal direction is in the direction of the nose and the temporal direction is away from the nose.

For both eyes, the area of binocular vision is approximately 60° to the left and right of the central line of the head and approximately 46° up, and 67° down. The monocular field of view for both eyes (total field of view) is approximately 95° to the left and right of the central line of the head and approximately 46° up, and 67° down. Table 9.1 provides an summary of the limits, while Figure 9.2 shows the shape and extent of the different types of field of view for both eyes.

	Temporally	Nasally	Up	Down
Single Eye (monocular)	95°	60°	46°	67°
	Left	Right	Up	Down
Both eyes (binocular)	60°	60°	46°	67°
Both eyes (monocular)	95°	95°	46°	67°

Table 9.1: The field of view boundaries for a single eye, and for binocular and monocular vision for both eyes.

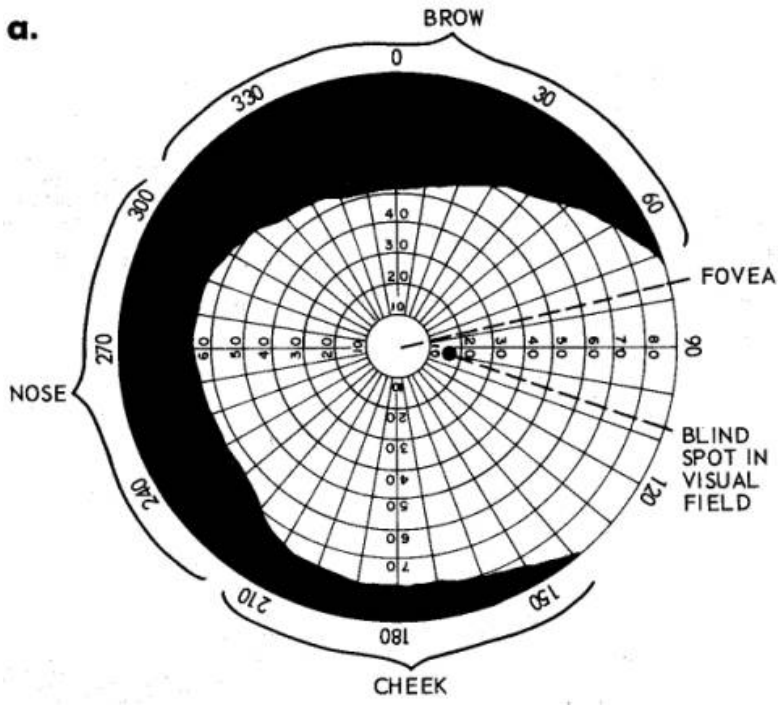


Figure 9.1: Field of view for a single eye (right eye, in this case), according to [PW73].

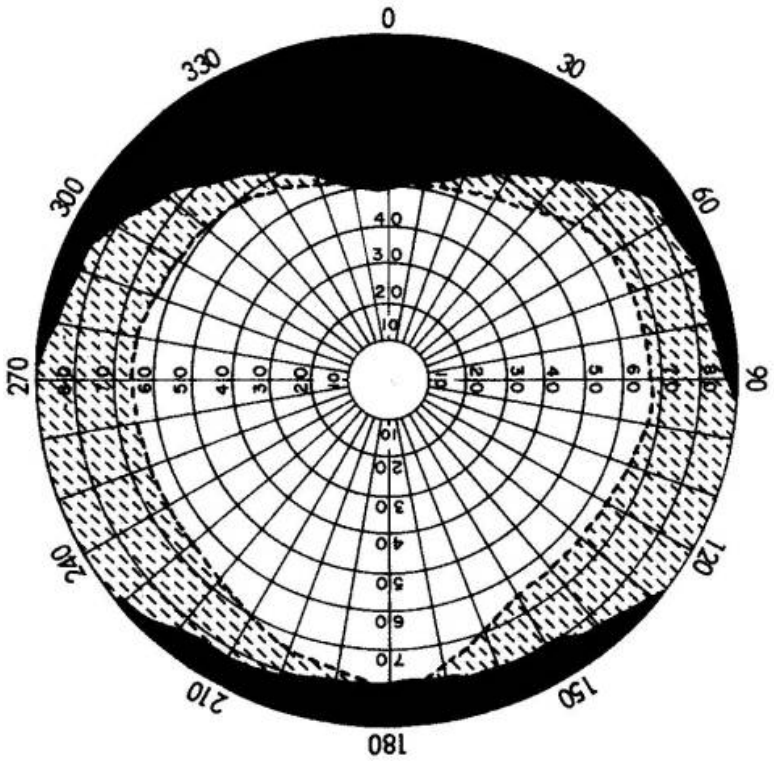


Figure 9.2: Field of view for both eyes. The white central area shows the binocular field of view, while the dashed area shows the areas only visible by at least one of the eyes. Image from [PW73].

9.1.2 Eye and Head Movements

In addition to the field of view of the eyes themselves, the overall field of view is affected by an additional factor —- the movement of the eyes and of the head. The eyes can move up to 74° in their sockets and the head can move up to 74°. This extends the potential field of view of a stationary person very significantly. The limits of combined head and eye movements, including direction, and their effects on the field of view are summarised in Table 9.2.

	Horizontal Limits		Vertical Limits	
	Left	Right	Up	Down
Maximum Head Movement	72°	72°	80°	90°
	Temporal (Ambinocular)	Nasal (Binocular)	Up	Down
Maximum Eye Movement	74°	55°	48°	66°
Maximum Range of Fixation	146°	127°	128°	156°
Peripheral FOV (from point of fixation)	91°	∼ 5°	18°	16°
Maximum Total FOV (from central body line)	237°	132°	146°	172°

Table 9.2: The various maxima relating to the field of view and how it is affected by head and eye movement. Values from [PW73].

While moving our eyes and head significantly extends where we can look, there is a temporal penalty due to the fact that moving muscles takes time. This is especially true for moving the head. When moving the eyes alone, the maximum angular velocity of the movement can be greater than $800^\circ/s$. Head movements are much slower, generally in the region of $100^\circ/s$. Moreover, when both the head and the eyes contribute to the gaze shift, the angular speed of the eye movement decreases as the head contribution increases. For angularly large gaze shifts ($> 50^\circ$) the maximum gaze velocity is approximately $400^\circ/s$ ($\sim 100^\circ/s$ head velocity and $\sim 300^\circ/s$ eye velocity) [Fre08].

Gaze shifts up to 20° tend to have minimum head movement. Larger movements tend to be comparatively slower with an increasing contribution of the head as the angular size of the gaze shift increases. For gaze shifts where there is minimal head movement contribution ($< 20^\circ$), the gaze shift is generally completed within 30–50 ms [Fre08].

9.1.3 Useful Field of View

Useful field of view (UFOV) is defined by Ball et al. as the "visual area within which useful information can be acquired without eye and head movements" [BRB90]. In the initial versions of the UFOV tests, it iwas generally measured as a radius in degrees, where the result represented the size of the visual field, in which the person can find a peripheral target with 50% accuracy. The size of UFOV, when measured, was between 5° and 35°.

However, it was later determined that the size of UFOV is affected by both the noticeability of the targets as well as the duration of the stimulus. Therefore, it is possible to detect more noticeable targets are at larger eccentricities or with shorter exposure times [EV+05]. Current versions of the UFOV test use the minimum stimulus duration at which the test subject has 75% performance accuracy for each of the subtasks as the defining measure [Ows13].

The main lesson to be learnt from UFOV related research is that the usable field of view of a person is affected not only by their age and the health of their visual system, but also by additional factors. These mainly consist of the presence of distractors, the attention of the person being divided and the difficulty of the task being performed [BRB90].

9.1.4 Visual Acuity

Visual acuity is a measure that corresponds to the eye's ability to resolve fine detail, essentially representing the minimum spatial frequency the eye can resolve. This is distinct from contrast sensitivity. Visual acuity is generally measured using a test that consist of high contrast shapes, usually black letters on white background. However, objects in everyday situations tend to have varying amounts of contrast relative to their surroundings. Grating patterns are used to measure contrast sensitivity at different spatial frequencies.

There are multiple related but distinct types of visual acuity. Their description and categorisation presented here is based on Katz and Kruger [KK06], and Shifman [Sch90]. Sometimes, the *absolute threshold* (aka. minimum visible acuity) is also included as one of them. The absolute threshold is defined as the minimum amount of light that needs to reach the retina to be detected. The luminosity of the object and the area of the retina illuminated by the object are used to compute the threshold. Aside from the *absolute threshold*, five types of acuity are generally recognised.

Detection acuity (aka. minimum perceptible acuity) describes the ability to detect an object within the visual field. This type of acuity is mostly dependent on brightness sensitivity of the eye and contrast of the object against its background. The visual angle can be as low as as 0.5 arc seconds, given high enough contrast against background.

Localisation acuity (aka. Vernier acuity) refers to the capacity to detect displacement or misalignment within a line. It is a form of hyper-acuity in that its minimum visual angle can be as low as 3-5 arc-seconds.

Resolution acuity (aka. minimum separable acuity) is the ability to see a separation between objects. The most common tests for this type of visual acuity are the two-point resolution test, Landolt C test and grating tests. Discrimination of two separate objects with a 1 arc-minute gap between them is referred to as having acuity of 1.0. Being able to discriminate objects with a gap of 0.5 arc-minute corresponds to acuity of 2.0.

Recognition acuity (aka. minimum legible acuity) is the most commonly known type of visual acuity and concentrates on the ability to identify the target objects within the field of vision. The best known tests for this type of acuity is the Snellen letter test or the logMAR test. A 20/20 Snellen fraction and 0 logMAR score correspond to 1 arc-minute acuity.

Dynamic acuity refers to the detection and location of a moving object. This acuity type depends on the size of the object and its angular velocity.

For the purpose of this section, and the thesis in general, acuity refers to *resolution acuity*, unless otherwise specified. The primary reason for using *resolution acuity* is that it allows direct comparison and relation to displays and pixels.

Estimating Visual Acuity A number of tests for measuring visual acuity are available. Since the primary focus of this appendix is on *resolution acuity*, the most practical way to estimate the maximum acuity for a person is to administer a Landolt C or a grating acuity test. However, in situations, where it is not possible, known results of *recognition acuity* tests may be used for rough estimation. Table 9.3 may be used as a guide for translating between the results of the different available tests.

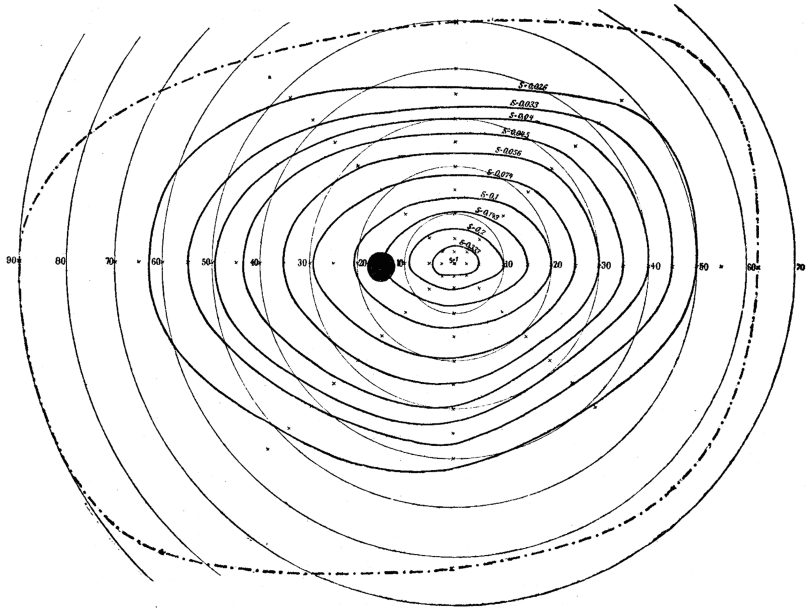


Figure 9.3: Visual acuity distribution within the left eye, shown as surface cutting vertically through the eye (the fixation point of the eye is the centre of the image). Image reproduced from [Wer94].

9.1.5 Depth of Field

The depth of field describes the distance between the nearest and furthest point that are in focus when the eye is focused at a certain distance. Depth of field is usually expressed in diopters, which are a unit of optical power, or as hyperfocal distance, which is the distance to the closest point still in focus if the eye is focused at infinity.

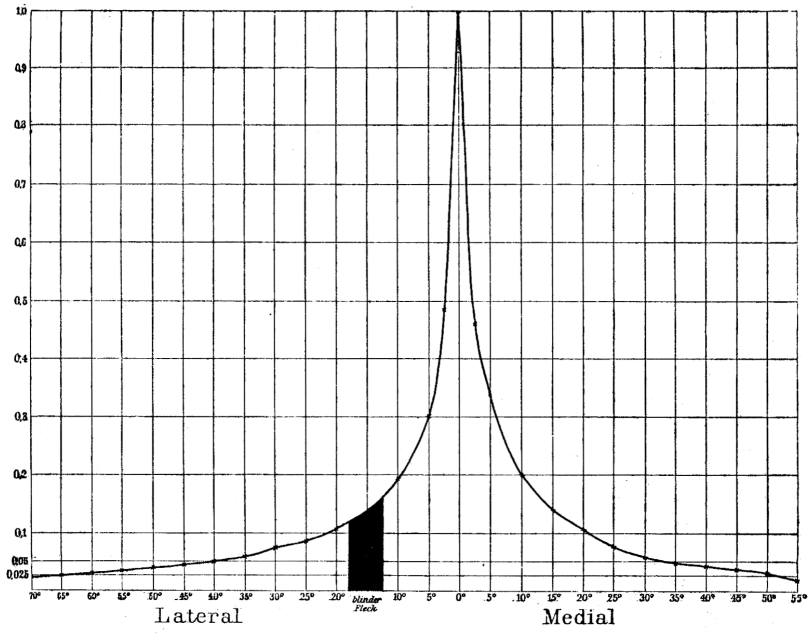


Figure 9.4: Visual acuity distribution in the left eye, shown as a cut through the surface in Figure 9.3 such that the plane is parallel to the ground if the person is sitting and their eye is pointed forward. Image reproduced from [Wer94].

The most significant factor affecting depth of field is the size of the pupil. Table 9.4 shows the depth of field for various pupil sizes when the eye is focused at several typical interaction distances. The resolving power in diopters (expressed as half of the total depth of field) and the hyperfocal distance of an eye with a specific pupil size is also included to enable calculation of the depth of field at arbitrary focal distances. The depth of field values were computed using base data from Campbell [Cam57].

Computing Depth of Field Before the depth of field can be computed, the size of the pupil has to be estimated. In 2012, Watson et al. [WY12] presented a unified formula for calculating pupil size under light adapted conditions. For ease of use, the formula is replicated and described in detail in Section 9.3.2.

Once the pupil diameter has been estimated, it is possible to estimate the depth of field. Campbell [Cam57] extensively studied the depth of field of the human eye under a variety of conditions and showed a relationship between the pupil size and depth of field. Unfortunately, Campbell [Cam57] only provided the data in a summary table rather than with a formula that fits his results. Therefore, the data is relatively sparse. Table 9.4 only shows results for the data as shown in Campbell's paper. However, if you use Watson et al.'s formula for the pupil size estimation described in Section 9.3.2 of the appendix, the result will be more granular. In order to enable estimates for depth of field outside the data presented in table 9.4, Formula 9.1 below can be used. The formula is based on a polynomial regression performed on the data from Campbell [Cam57] and accounts for over 99% of the variance.

$$\mathrm{D} = \frac{1}{-0.0541p^2 + 1.2709p - 0.0376} \tag{9.1}$$

Snellen Fractions			Min. Angle of Resolution		Cycles	Visual Angle	Points
feet	metres	dec. eq.	arc-minutes	logMAR	per Degree	degrees	per Degree
20/10	6/3	2.00	0.50	-0.30	60	0.0083	119.72
20/12.5	6/3.75	1.60	0.63	-0.20	48	0.0100	95.09
20/16	6/4.8	1.25	0.80	-0.10	38	0.0132	75.54
20/20	6/6	1.00	1.00	0.00	30	0.0167	60.00
20/25	6/7.5	0.80	1.25	0.10	24	0.0210	47.66
20/32	6/9.5	0.63	1.60	0.20	19	0.0264	37.86
20/40	6/12	0.50	2.00	0.30	15	0.0333	30.07
20/50	6/15	0.40	2.50	0.40	12	0.0419	23.89
20/63	6/18.9	0.32	3.15	0.50	9	0.0527	18.97
20/80	6/24	0.25	4.00	0.60	8	0.0664	15.07
20/100	6/30	0.20	5.00	0.70	6	0.835	11.97
20/125	6/37.5	0.16	6.25	0.80	5	0.1052	9.51
20/160	6/48	0.13	8.00	0.90	4	0.1324	7.55
20/200	6/60	0.10	10.00	1.00	3	0.1667	6.00

Table 9.3: Table of equivalence of currently used units for measuring visual acuity (values gathered from [Hol97] and[Vis88] or computed manually). The values in the last two columns were rounded to four and two decimal places, respectively.

where D is the half the depth of field in diopters and p is the pupil size in millimetres. Note that while this formula allows for more accurate computation of the depth of field, unless you use accurate input values, the result may be spuriously accurate. The formula was derived from the relationship between the pupil size and hyperfocal distance as seen in Figure 9.5.

There is a very useful relationship between diopters and focal distance, where the optical power in diopters is the inverse of the focal distance in metres. This means that it is trivial to determine the optical power of the eye required for it to focus at a specific distance. Hyperfocal distance is a distance, from which everything appears in focus if the eye (or a lens) is focussed on infinity. Due to the inverse relationship between optical power and focal distance, it can also be used as a measure of optical power of a lens (or the eye). Since the hyperfocal distance effectively forms the near boundary of the depth of field at infinity, it corresponds to one half of the total depth of field. This means that when it is converted into diopters using the inverse relationship of focal distance and optical power, the result is half of the depth of field of the eye (or a lens).

Since the optical power does not change with focal distance, it is possible to use the optical power to compute the depth of field at any focal distance. Moreover, since the optical power is expressed as half of depth of field, computing the boundaries of the depth of field for a specific focal distance is quite simple. Formulas 9.2 and 9.3 can be used to compute the near and far boundary, respectively.

$$B_{\text{near}} = \frac{f}{1 - Df} = \frac{f}{1 - \frac{f}{-0.0541p^2 + 1.2709p - 0.0376}} \tag{9.2}$$

$$B_{\text{far}} = \frac{f}{1 + Df} = \frac{f}{1 + \frac{f}{-0.0541p^2 + 1.2709p - 0.0376}} \tag{9.3}$$

where B_{near} and B_{far} are the near and far boundaries of the depth of field in metres, respectively; D is half the depth of field in diopters and f is the focal distance of the eye. Important note: When $\frac{1}{f} - D < 0$ for the near boundary or $\frac{1}{f} + D < 0$ for the far boundary, the results is *infinity*.

The presented formulas are based on results from [Cam57] and generally represent the best case scenario. However, other researchers have investigated additional influences on the depth of field

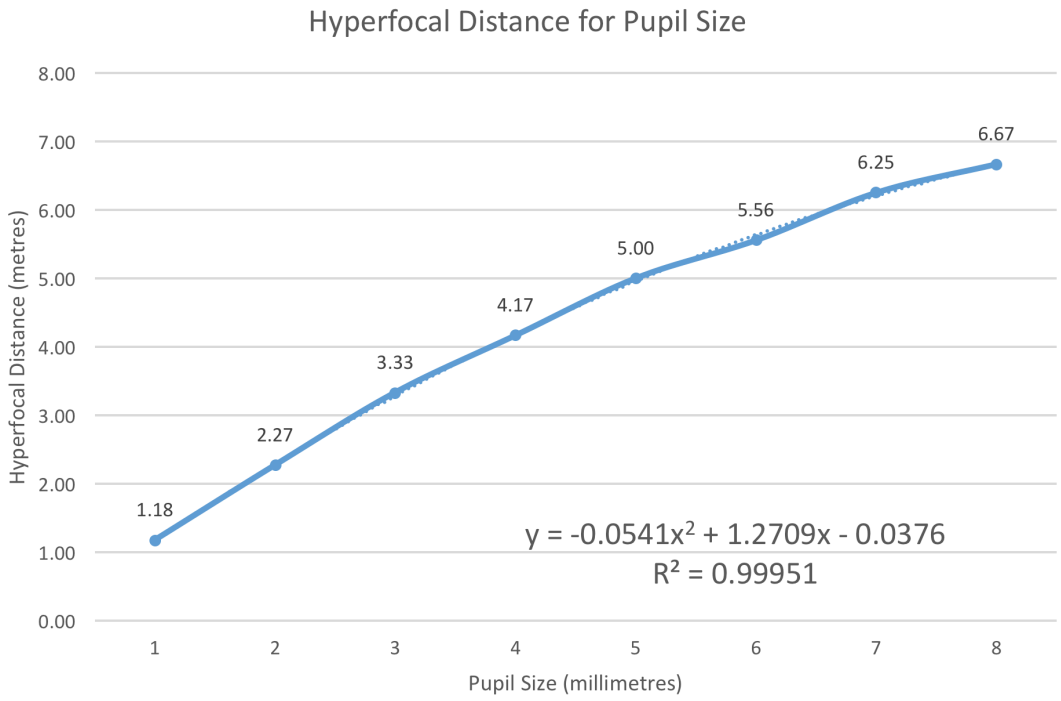


Figure 9.5: Hyperlocal distance for specific pupil sizes (using data from [Cam57]). Displayed formula is a polynomial regression, which accounts for 99.9% of the variance ($R^2 = 0.99951$).

Pupil ⌀	DOF (diopter)	Hyperfocal Distance	Range @0.35m (phone/tablet)		Range @0.5m (laptop)		Range @0.65m (desktop)		Range @2.5m (TV)	
			Near	Far	Near	Far	Near	Far	Near	Far
1	±0.85	1.18	0.270	0.498	0.351	0.870	0.419	1.455	0.799	∞
2	±0.44	2.27	0.303	0.414	0.410	0.641	0.505	0.911	1.189	∞
3	±0.3	3.33	0.317	0.391	0.435	0.588	0.544	0.807	1.429	10.000
4	±0.24	4.17	0.323	0.382	0.446	0.568	0.562	0.770	1.563	6.250
5	±0.2	5.00	0.327	0.376	0.455	0.556	0.575	0.747	1.667	5.000
6	±0.18	5.56	0.329	0.374	0.459	0.549	0.582	0.736	1.724	4.545
7	±0.16	6.25	0.331	0.371	0.463	0.543	0.589	0.725	1.786	4.167
8	±0.15	6.67	0.333	0.369	0.465	0.541	0.592	0.720	1.818	4.000

Table 9.4: The depth of field, hyperlocal distance and range of focus for various pupil sizes. The pupil sizes are expressed as the diameter of the pupil in millimetres. Depth of field is expressed as half the depth in diopters. The hyperfocal distance and focal ranges are in metres. The four distances, for which the range of focus has been calculated, represent typical distances at which various devices are used ([FK+07; LR+14]). The basis for the calculations is the data from Campbell [Cam57].

of the human eye, so if the target interaction scenario requires more specific results, other sources, including the following should be consulted. Atchison et al. [ACW97] conducted experiments to examine the interplay of target size, contrast and pupil size on DOF. Marcos et al. [MMN99] compared objective DOF measurements (as used here) with subjective values. Wang et al. [WC04] investigated DOF changes outside of the foveal region of the eye. More recently, Wang et al. [WC06] compiled a review of other this and other DOF research.

9.1.6 Accommodation

Accommodation is the eye's ability to alter the shape of its lens to maintain focus. It is included here because the accommodation has a possible influence on the timings of interactions. In terms of time, the accommodation process has approximately a 300ms lag and it can take up to several seconds (although times around 0.5 s are more common) to stabilise the eye's focus at a new distance, depending on the magnitude of change [CW60; Cha08]. This means that interactions that rely on visual switches between surfaces with large depth disparities should be carefully considered, if the visual switching is time-sensitive.

Additionally, the eye's ability to focus at different distances changes with age. Table 9.5 provides an overview of predicted minimum accommodation distances for people at different ages, based on data collected by Duane [Dua22]. The main trend is that up to approximately the late thirties, the minimum focal distances are relatively stable in the region of 10–20 cm. However, in their early forties, people's minimum focal distances start to deteriorate relatively fast, until another relative plateau is reached towards the their early sixties. However, at this stage, interactions relying on being able to clearly see visual content at short distances become difficult or possibly impractical without the use of corrective aids such as glasses or contact lenses.

9.2 Considerations

So far, this appendix has provided a brief, non-exhaustive sampling of findings and methods from existing literature with a particular focus on visual acuity and related topics. At times the findings have been augmented with methods to simplify finding values for specific parameters relating to human vision and display characteristics. This section distills some of the presented knowledge into considerations for designers of future systems that include displays and other visual elements. Additionally, this section also highlights considerations that should be taken into account when designing interactions with a spatial element.

In order to simplify the presentation of the findings, a set of prototypical displays will be used throughout the rest of the appendix. Table 9.6 summarises their basic properties. The prototypical displays have been chosen to represent a set of devices that people commonly interact with, and to provide a range of physical sizes and pixel densities to better demonstrate some of their effects.

9.2.1 Display Viewing Distance

In any system that includes spatial interactions, especially when using distance or position relationships, it is likely that the viewing distance of displays will be variable. However, even in systems with mostly static interaction distances (e.g. *DiffDisplays* in Chapter 7), the viewing distance to displays still affects the apparent size and pixel density of the displays.

It is advisable is to use displays that have a pixel density close to the person's visual acuity, when viewed at the desired distance. Table 9.7 shows the recommended minimum viewing distances for the prototypical displays for a range of visual acuity values. The minimum viewing distance is defined as the distance, from which the apparent pixel density of display will approximately match the person's visual acuity. While a display with sufficient pixel density to match the person's acuity is desirable, it is not necessary for it to have a pixel density significantly higher than that. This is because at pixel densities beyond a person's visual acuity, the person will not be able to perceive any additional detail. For relatively static systems without large changes in distance to the display, the apparent size of the display is of primary concern as the display resolution can be easily optimised (within technological constraints).

However, if the viewing distance to the display is highly variable, both the apparent pixel density and the apparent size of the display will be equally variable. The apparent pixel density of a display

Resolving Power (diopters)								Minimum Focusing Distance (metres)							
Age	Best	Mean	Worst	Age	Best	Mean	Worst	Age	Best	Mean	Worst	Age	Best	Mean	Worst
8	16.1	13.8	11.6	41	7.5	5.4	3	8	0.06	0.07	0.09	41	0.13	0.19	0.33
9	15.9	13.6	11.4	42	7.1	5	2.7	9	0.06	0.07	0.09	42	0.14	0.20	0.37
10	15.7	13.4	11.1	43	6.7	4.5	2.3	10	0.06	0.07	0.09	43	0.15	0.22	0.43
11	15.5	13.2	10.9	44	6.3	4	2.1	11	0.06	0.08	0.09	44	0.16	0.25	0.48
12	15.2	12.9	10.7	45	5.9	3.6	1.9	12	0.07	0.08	0.09	45	0.17	0.28	0.53
13	15	12.7	10.5	46	5.5	3.1	1.7	13	0.07	0.08	0.10	46	0.18	0.32	0.59
14	14.8	12.5	10.3	47	5	2.7	1.4	14	0.07	0.08	0.10	47	0.20	0.37	0.71
15	14.5	12.3	10.1	48	4.5	2.3	1.2	15	0.07	0.08	0.10	48	0.22	0.43	0.83
16	14.3	12	9.8	49	4	2.1	1.1	16	0.07	0.08	0.10	49	0.25	0.48	0.91
17	14.1	11.8	9.6	50	3.2	1.9	1	17	0.07	0.08	0.10	50	0.31	0.53	1.00
18	13.9	11.6	9.4	51	2.6	1.7	0.9	18	0.07	0.09	0.11	51	0.38	0.59	1.11
19	13.6	11.4	9.2	52	2.2	1.6	0.9	19	0.07	0.09	0.11	52	0.45	0.63	1.11
20	13.4	11.1	8.9	53	2.1	1.5	0.9	20	0.07	0.09	0.11	53	0.48	0.67	1.11
21	13.1	10.9	8.7	54	2	1.4	0.8	21	0.08	0.09	0.11	54	0.50	0.71	1.25
22	12.9	10.7	8.5	55	1.9	1.3	0.8	22	0.08	0.09	0.12	55	0.53	0.77	1.25
23	12.6	10.5	8.3	56	1.8	1.3	0.8	23	0.08	0.10	0.12	56	0.56	0.77	1.25
24	12.4	10.2	8	57	1.8	1.3	0.8	24	0.08	0.10	0.13	57	0.56	0.77	1.25
25	12.2	9.9	7.8	58	1.8	1.3	0.7	25	0.08	0.10	0.13	58	0.56	0.77	1.43
26	11.9	9.7	7.5	59	1.7	1.2	0.7	26	0.08	0.10	0.13	59	0.59	0.83	1.43
27	11.6	9.5	7.2	60	1.7	1.2	0.7	27	0.09	0.11	0.14	60	0.59	0.83	1.43
28	11.3	9.2	7	61	1.7	1.2	0.6	28	0.09	0.11	0.14	61	0.59	0.83	1.67
29	11	9	6.8	62	1.6	1.2	0.6	29	0.09	0.11	0.15	62	0.63	0.83	1.67
30	10.8	8.7	6.5	63	1.6	1.1	0.6	30	0.09	0.11	0.15	63	0.63	0.91	1.67
31	10.5	8.4	6.2	64	1.6	1.1	0.6	31	0.10	0.12	0.16	64	0.63	0.91	1.67
32	10.2	8.1	6	65	1.6	1.09	0.6	32	0.10	0.12	0.17	65	0.63	0.92	1.67
33	9.8	7.9	5.8	66	1.6	1.08	0.6	33	0.10	0.13	0.17	66	0.63	0.93	1.67
34	9.5	7.6	5.5	67	1.6	1.06	0.6	34	0.11	0.13	0.18	67	0.63	0.94	1.67
35	9.3	7.3	5.2	68	1.6	1.05	0.6	35	0.11	0.14	0.19	68	0.63	0.95	1.67
36	9	7	4.9	69	1.6	1.04	0.6	36	0.11	0.14	0.20	69	0.63	0.96	1.67
37	8.8	6.7	4.5	70	1.6	1.03	0.6	37	0.11	0.15	0.22	70	0.63	0.98	1.67
38	8.5	6.4	4.1	71	1.6	1.01	0.6	38	0.12	0.16	0.24	71	0.63	0.99	1.67
39	8.2	6.1	3.7	72	1.6	1	0.6	39	0.12	0.16	0.27	72	0.63	1.00	1.67
40	7.9	5.8	3.4					40	0.13	0.17	0.29				

Table 9.5: Predicted limits to resolving power and minimum focussing distances for ages 8–72 based on data from Duane [Dua22]. Values for resolving power were collected from [Dua22] and minimum focussing distance was computed by the thesis author.

Display Type	Aspect Ratio Horiz.	Vert.	Diagonal (inches)	Dimensions (cm) Width	Height	Resolution (pixels) Horiz.	Vert.	Pixel Density (ppi)
Mobile Phone	16	9	4.7	10.1	6.3	1334	750	326
Tablet	4	3	9.7	19.7	14.8	2048	1536	264
Laptop	16	10	15.4	33.2	20.7	2880	1800	221
Desktop	16	9	27	59.8	33.6	3840	2160	163
TV (1080p)	16	9	50	110.7	62.3	1920	1080	44
TV (UDH)	16	9	50	110.7	62.3	3840	2160	88

Table 9.6: Physical properties of prototypical examples of displays.

is directly proportional to the viewing distance, while apparent size is inversely proportional. That is, as viewing distance increases, the apparent pixel density of the display increases, while the apparent size decreases. Therefore, in interactive systems, where large changes in viewing distance are expected, both the physical size of the display as well as its resolution need to be balanced.

An additional concern when designing distance-based interactions with displays is the minimum focusing distance of the user. Table 9.5 shows predicted ranges of minimum focusing distance for people of various ages, who do not wear glasses, contact lenses or use other corrective visual aids. For people who use visual aids, their minimum and maximum focussing distance is likely to be determined by the corrective visual aid. For example, reading glasses tend to be optimised for 40cm viewing distance [Tho98].

Display Type	Minimum optimal viewing distance at Snellen fractions (cm) 20/10	20/12.5	20/16	20/20	20/25	20/32	20/40	20/50	20/63	20/80
Mobile Phone	53.6	42.6	33.5	26.8	21.5	16.8	13.4	10.7	8.5	6.7
Tablet	66.2	52.5	41.4	33.1	26.5	20.7	16.5	13.2	10.5	8.3
Laptop	79.2	62.8	49.5	39.6	31.7	24.7	19.8	15.8	12.6	9.9
Desktop	107.0	84.9	66.9	53.5	42.8	33.4	26.8	21.4	17.0	13.4
TV (1080p)	396.4	314.6	247.7	198.2	158.5	123.9	99.1	79.3	62.9	49.5
TV (UDH)	198.2	157.3	123.9	99.1	79.3	61.9	49.5	39.6	31.5	24.8

Table 9.7: Minimum optimal viewing distance for different visual acuities expressed as Snellen fractions.

9.2.2 Display Viewing Angles

The prototype systems in Chapter 5 demonstrated that the limited viewing angles of TN LCD displays can be leveraged for creating novel interactions. However, in general use these properties tend to manifest as limitations rather than advantages. Different manufacturing methods of LCD displays lead to differing visual stability of LCD display at various viewing angles. However, the specifics of colour, luminance and contrast changes in different types of displays is beyond the scope of this appendix. We only note that system designers should consider the different visual properties of available displays when designing systems as e.g. LCD displays exhibit different characteristics compared to projection based displays or Cathode Ray Tube (CRT) displays.

The main point of consideration regarding display viewing angles relates to how it affects the distribution of pixels on the display. Essentially viewing a display at an angle other than perpendicular to the display leads to *resolution compression* because the visual angle occupied by the display decreases even as the distance to the display remains the same. In order to quantify how much the physical resolution of the display is compressed, apparent pixel density should be used.

$$\text{AngularSize} = \tan^{-1}\left(\frac{x\cos\beta}{d - x\sin\beta}\right) + \tan^{-1}\left(\frac{x\cos\beta}{d + x\sin\beta}\right) \tag{9.4}$$

The apparent pixel density can be computed using Formula 9.4, where x is half the length of the display in the dimension used for computation, β is the angle of view (defined as the angular difference from a line perpendicular to the display in the dimension used for computation) and d is the distance of the viewer from the centre of the display. Note that the angular difference β is always positive (i.e. 30° to the left of the display is equivalent to 30° to the right of the display) and cannot be larger than 90° as the back of the display is not visible. Figure 9.6 shows several examples.

Once the angular size of the display along a specific axis has been computed, simply dividing the pixel resolution along the chosen axis by the computed angular size will yield the apparent pixel density. This could be applied in a similar fashion as the considerations regarding viewing distance, especially with relatively static spatial arrangements. If a person is likely to see a particular display from an angle not perpendicular to the display, the pixel density of the display considered when determining optimal viewing distance should be based on the apparent pixel density, taking the viewing angle into account.

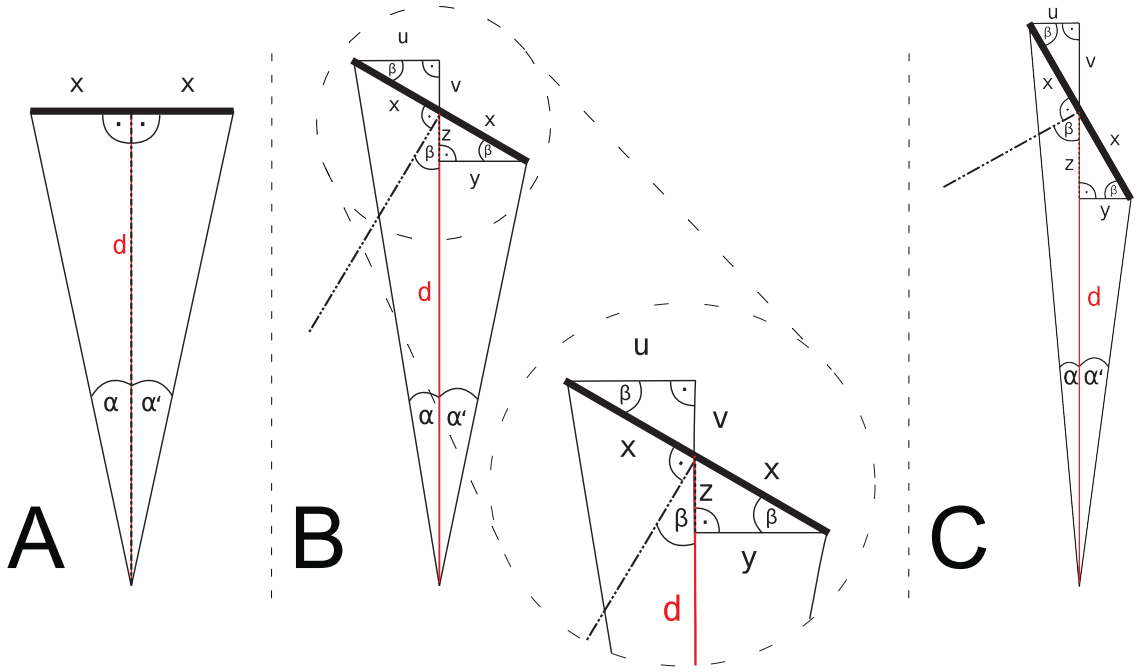


Figure 9.6: Angular size of a display at any viewing angle. Subfigure *A* shows an arrangement where the observer is perpendicular to the display. Subfigures *B* and *C* shows an arrangement with the observer at 30° and 60° from the display, respectively. Inset in Subfigure *B* is a magnified view of the most significant part of the subfigure to provide a more easily readable view of the geometrical relationships. For definitions of the notation and related calculations, see Appendix 9.3.1.

9.2.3 Display Size and Positioning

Apart from pixel density, the other significant factor that could influence the choice of a display is it's apparent size from the viewpoint of the person looking at it. Table 9.8 shows the angular size of the prototypical display at a number of distances. The factors to consider here are whether the

application scenario requires content to be distributed across the entire display or only a part of it. If the person needs to simultaneously perceive content at different parts of the display, using a measure such as the expected useful field of view may be desirable (see Section 9.1.3 for more details). However, another simple heuristic can also be used. The display should probably occupy an area that covers less than the radius of the total area in which the person can see with sufficient visual acuity for the required scenario. This angular radius can be determined by reading values from Figures 9.4 and 9.3 and adding approximately 20° to it. This is because if the eye-movement required is less than approximately <20°, it is likely a head movement is not going to take place and the eye movement will be finished within 30-50 ms [Fre08].

	Horizontal angular size (degrees) at distance (cm)											
Display Type	25	50	75	100	150	200	250	300	350	400	450	500
Mobile Phone	22.9	11.6	7.7	5.8	3.9	2.9	2.3	1.9	1.7	1.4	1.3	1.2
Tables	43.0	22.3	15.0	11.3	7.5	5.6	4.5	3.8	3.2	2.8	2.5	2.3
Laptop	67.1	36.7	24.9	18.8	12.6	9.5	7.6	6.3	5.4	4.7	4.2	3.8
Desktop	100.2	61.7	43.5	33.3	22.5	17.0	13.6	11.4	9.8	8.5	7.6	6.8
TV (1080p)	131.4	95.8	72.8	57.9	40.5	30.9	25.0	20.9	18.0	15.8	14.0	12.6
TV (UDH)	131.4	95.8	72.8	57.9	40.5	30.9	25.0	20.9	18.0	15.8	14.0	12.6

Table 9.8: Angular size of prototypical displays at different distances for 20/20 visual acuity, assuming the observer is perpendicular to the display.

When viewing displays at angles other than approximately perpendicular to the display, there is an additional consideration related to the position of the display. The depth of field and the distance to the display determine whether the entire display will be in focus or not. If the display is relatively close to the person, relative to its physical size, it is likely that if viewed at relatively acute angles to the display, the depth of field of the person may not be sufficient to cover the entire display. Refer to Table 9.4 for examples of the near and far depth of field distances for eyes focused at common interaction distances. If the distance between the closest and furthest parts of the display is larger than the person's depth of field at that distance, the person will need to refocus their eyes when looking at different parts of the display.

The depth of field constraint also applies in multi-display scenarios. The depth disparity of the displays will influence the cost of switching between the displays. Rashid et al. [RNQ12] introduced a visual arrangement of multi-display configurations, based on visual field and depth continuity. Following their taxonomy, for displays with only visual field discontinuities, as long as the approximate viewing distance is maintained, the cost of switching is likely to be relatively small, at least in terms of the person's vision. This is because only an eye movement is likely required.

However, if there is a significant depth discontinuity between the displays, the switch between them may require a change in the accommodation of the eyes. If the depth discontinuity between the displays is larger than the near or far boundary (depending on the direction of the discontinuity) of the depth of field, a change in accommodation of the eye will likely be required. The re-focusing of the eyes can take over a second to complete [CW60; Cha08], which means that multi-display configurations with large depth disparities should be carefully considered if the switches between the displays are time-sensitive.

9.3 Calculations

This section contains the calculations used to determine some of the original formulas used in this appendix or to provide additional formulas that can be used for coarse estimations where more precise methods are not available.

9.3.1 Angular Size

This section presents the derivation of a formula for computing the angular size (in a given dimension) of a display viewed from a specific distance. The formula takes into account the viewing angle when the display is viewed at an angle rather than straight on.

Proof. Let $u, v, x, y, z, d, \alpha, \alpha' \beta$ be according to Figure 9.6, where x is $\frac{1}{2}$ of the length of one display dimension (width, height), d is the viewing distance of the observer from the display, and β is the viewing angle of the observer towards the display. So,

$$\begin{aligned} u &= x\cos\beta, && \text{(trigonometry)} \\ v &= x\sin\beta, \\ y &= x\cos\beta, \\ z &= x\sin\beta, \end{aligned}$$

$$\begin{aligned} \tan\alpha &= \frac{u}{d-v} && \text{(trigonometry)} \\ &= \frac{x\cos\beta}{d-(x\sin\beta)} && \text{(substituting variables)} \end{aligned}$$

$$\begin{aligned} \tan\alpha' &= \frac{y}{d+z} && \text{(trigonometry)} \\ &= \frac{x\cos\beta}{d+(x\sin\beta)} && \text{(substituting variables)} \end{aligned}$$

$$\begin{aligned} \text{AngularSize} &= \alpha + \alpha' \\ &= \tan^{-1}\left(\frac{x\cos\beta}{d-x\sin\beta}\right) + \tan^{-1}\left(\frac{x\cos\beta}{d+x\sin\beta}\right) && \text{(substituting variables)} \end{aligned}$$ □

Proof. For $\beta = 0^\circ$ (which should be equivalent to the rightmost example in Figure 9.6), $\sin\beta = 0$, $\cos\beta = 1$

$$\begin{aligned} \text{AngularSize} &= \tan^{-1}\left(\frac{x\cos\beta}{d-x\sin\beta}\right) + \tan^{-1}\left(\frac{x\cos\beta}{d+x\sin\beta}\right) \\ &= \tan^{-1}\left(\frac{x\times 1}{d-x\times 0}\right) + \tan^{-1}\left(\frac{x\times 1}{d+x\times 0}\right) \\ &= \tan^{-1}\left(\frac{x}{d}\right) + \tan^{-1}\left(\frac{x}{d}\right) && \text{equiv. to rightmost example in Figure 9.6} \end{aligned}$$ □

Proof. For $\beta = 90^\circ$ (for a situation where the observer is at 90° from the display), $\sin\beta = 1$, $\cos\beta = 0$,

$$
\begin{aligned}
\text{AngularSize} &= \tan^{-1}\left(\frac{x\cos\beta}{d - x\sin\beta}\right) + \tan^{-1}\left(\frac{x\cos\beta}{d + x\sin\beta}\right) \\
&= \tan^{-1}\left(\frac{x \times 0}{d - x \times 1}\right) + \tan^{-1}\left(\frac{x \times 0}{d + x \times 1}\right) && \text{(substituting variables)} \\
&= \tan^{-1}\left(\frac{0}{d - x}\right) + \tan^{-1}\left(\frac{0}{d + x}\right) \\
&= \tan^{-1}(0) + \tan^{-1}(0) \\
&= 0 + 0 \\
&= 0
\end{aligned}
$$

□

9.3.2 Computing Pupil Size

Watson et al. introduced a unified formula for calculating pupil size under light adapted conditions based on a number of previously published models. The formula (in simplified form) from Watson et al.[WY12] , as well as the parameters required are summarised below. Note that this formula will produce useful results for most interactional situations. However, it will not produce good results in situations where the eyes will not be light adjusted. The eyes are unlikely to be light adapted in the following situations:

- During the first few minutes of interaction if the interaction environment is located in an area with very different lighting conditions than the person was exposed to previously. Because the person will have moved from a very brightly lit environment to a low light situation (or vice versa), it will take some time for their eyes to adjust to the new conditions.
- If the interaction involves gaze shifts between very bright and very dim areas. For an extreme example, if a person in the cinema had to switch between watching a film on the screen and reading from a piece paper (passively lit by the screen).

In most common interaction scenarios, the lighting conditions will be such that after accounting for the initial adaptation period, the formula below will produce usable results.

In order to estimate the pupil size in a particular situation, the following parameters need to be known:

Age The age of the person. If using the simplified formulas for determining pupil size (9.7, 9.8) rather the full formula (all by [WY12]), it is enough to know whether the person above or below 20 years of age.

Number of Eyes that are adapted to the existing lighting conditions. This will generally be two in most interaction scenarios as vision will not be obstructed.

Luminance is the luminance within the adaptation field, ideally measured using a light-meter or photon-meter. If a light-meter or photon-meter cannot be used, Section 9.3.3 of this appendix can be used to produce a reasonable estimate.

Adaptation Field Area The area, which is used by the eyes to determine it's light/dark adaptation level. In an environment with little ambient light (e.g. watching the TV in the evening with no lights on), this will correspond to the angular area occupied by the display that is being interacted with. In an environment where ambient light is relatively bright, the value is the total area that the eye perceives light from. In every case, the value should be expressed as a radius of a circle with equivalent area. In cases where Section 9.3.3 was used to estimate the

luminance value, the methods there should also be used to provide a value for the adaptation field area.

Before calculating the pupil diameter, it is necessary to compute the effective corneal flux density using formula 9.6. Effective corneal flux density, where L is the luminance in cd/m^2, a is the field area in deg^2 and $M(e)$ is a function expressing the attenuation factor for light as a function of the number of eyes looking at the stimulus (defined in Formula 9.5).

Once the effective corneal flux has been computed, the result should be applied to either the full formula described in Watson et al. [WY12], or to the simplified formulas presented here. The simplified formulas are also replicated from Watson et al. [WY12] and only included for ease of use. If the person less than 20 years old, Formula 9.8 should be applied, and if they are more than 20 years old, Formula 9.7 should be used. Once the pupil size has been estimated, the result can be used for estimating the depth of field as described in Section 9.1.5.

$$\begin{aligned} M(1) &= 0.1 \\ M(2) &= 1 \end{aligned} \tag{9.5}$$

$$f = F^{0.41} = [LaM(e)]^{0.41} \tag{9.6}$$

$$D_U = \frac{18.5172 + 0.122165f - 0.105569y + 0.000138645fy}{2 + 0.0630635f} \tag{9.7}$$

$$D_U = \frac{16.4674 + \exp[-0.208269y \times (-3.96868 + 0.00521209f)] + 0.124857f}{2 + 0.0630635f} \tag{9.8}$$

9.3.3 Using a Camera to Estimate Luminance and Adapting Area

Estimating Luminance

The majority of the population does not own or have access to a lightmeter for accurate luminance estimation. Fortunately, analogue and digital cameras generally come with an integrated metering system for determining exposure settings. The metering system can be leveraged for estimating luminance. The accuracy of such estimation is likely to be lower than if a calibrated lightmeter was used, but for the purpose of estimating luminance of a given environment, it will give a reasonably accurate result. For a discussion on the accuracy of digital cameras as lightmeters, see [WG+07]. The formulas below are based on equations from [Lin].

Proof. Let a be the aperture value (the f-number), t be the shutter speed in seconds, i be the ISO, and L be the luminance in cd/m^2 and B be luminance in foot-Lamberts. Also, let EV be the exposure value, BV be the brightness value, and SV be the speed value. So,

$$\begin{aligned} \text{EV} &= \log_2(a^2) + \log_2(\frac{1}{t}), \\ \text{EV} &= \log_2(B) + \log_2(0.32i), \\ L &= 3.4262590996323B \end{aligned}$$

$$
\begin{aligned}
\mathrm{EV} &= \log_2(a^2) + \log_2(\frac{1}{t}) \\
\log_2(B) + \log_2(0.32i) &= \log_2(a^2) + \log_2(\frac{1}{t}) && \text{(substituting variables)} \\
\log_2(B) + \log_2(0.32i) &= \log_2(\frac{a^2}{t}) \\
\log_2(B) &= \log_2(\frac{a^2}{t}) - \log_2(0.32i) \\
\log_2(B) &= \log_2(\frac{a^2}{0.32ti}) \\
B &= \frac{a^2}{0.32ti} \\
3.4262590996323B &= 3.4262590996323\frac{a^2}{0.32ti} \\
L &= 3.4262590996323\frac{a^2}{0.32ti} && \text{(substituting variables)} \\
L &= 10.7070596863509\frac{a^2}{ti} && \square
\end{aligned}
$$

Estimating Adapting Area

In order to estimate the adapting area when using a camera to estimate luminance, we need to compute the field of view of the camera (expressed as an area), which will then be used to derive the diameter of a circle with the same area. Computing the field of view of a lens in a specific dimension can be done using the formula below (Formula 9.9):

$$\alpha = 2\tan^{-1}(\frac{d}{2f}) \tag{9.9}$$

where α is the field of view expressed as an angle, d is the length of the image sensor in a particular dimension and f is the focal length of the lens (when focussed at infinity).

In order to compute the diameter of the equivalent adapting area, the formula derived below can be used.

Proof. Let f be the focal length of the lens, h and v be the horizontal and vertical length of the image sensor in millimetres, and α and β be the horizontal and vertical angle of view. Also, let A be the adapting area and r be the radius of the adapting area expressed as a circle and d be the diameter of the same circle. So,

$$
\begin{aligned}
\alpha &= 2\tan^{-1}(\frac{h}{2f}), && \text{(horizontal FOV)} \\
\beta &= 2\tan^{-1}(\frac{v}{2f}), && \text{(vertical FOV)} \\
A &= \alpha\beta, && \text{(FOV of the camera as an area)} \\
A &= \pi r^2, \\
d &= 2r
\end{aligned}
$$

$$
\begin{aligned}
A &= \alpha\beta \\
A &= 4\tan^{-1}(\frac{h}{2f})\tan^{-1}(\frac{v}{2f}) && \text{(substituting variables)} \\
\pi r^2 &= 4\tan^{-1}(\frac{h}{2f})\tan^{-1}(\frac{v}{2f}) && \text{(substituting variables)} \\
r^2 &= \frac{4\tan^{-1}(\frac{h}{2f})\tan^{-1}(\frac{v}{2f})}{\pi} \\
r &= \sqrt{\frac{4\tan^{-1}(\frac{h}{2f})\tan^{-1}(\frac{v}{2f})}{\pi}} \\
2r &= 2\sqrt{\frac{4\tan^{-1}(\frac{h}{2f})\tan^{-1}(\frac{v}{2f})}{\pi}} \\
d &= 2\sqrt{\frac{4\tan^{-1}(\frac{h}{2f})\tan^{-1}(\frac{v}{2f})}{\pi}} && \square
\end{aligned}
$$

9.4 Conclusions

This appendix contributes and synthesises existing knowledge, understanding and models related to human vision relevant for designers of interactive systems that include spatial elements. Due to the breath of existing knowledge about human vision, the scope of this appendix is limited to visual acuity and aspects of vision that affect it. Additionally, the literature summarised in this appendix only includes the minimum amount of detail required to demonstrate reasons for the presented considerations. Researchers and system designers are encouraged to explore additional sources when informing their designs. This is especially the case with systems where a visually impaired person may interact with the system as visual impairments are unfortunately beyond the scope of this appendix.

These considerations are important for the designers of tracking technologies in Chapter 4. Similarly, aspects of the case studies presented in Chapters 5, 6, 7 can be understood in more depth with respect to the models and data on human vision, distance, viewing angles, outlined here.

The content presented in this appendix contains, or points to, data, understanding and knowledge for addressing the major concerns. One of the main outcomes of the analysis of existing systems in Chapter 3 was that researchers and designers of spatially-aware displays do not seem to frequently explicitly consider the interactive environment, in which the explored interactions takes place, or the visual specifics of the displays. While this is not entirely surprising given the specialised nature of research publications, a certain amount of explicit consideration would likely be beneficial. Additionally, it may also reveal further research opportunities.

Bibliography

[AA+96] Gregory D Abowd, Christopher G Atkeson et al. 'Cyberguide : A Mobile Context-Aware Tour Guide'. In: *Baltzer Journals* 3.September 1996 (1996), pp. 1–21. ISSN: 10220038. DOI: 10.1023/A:1019194325861.

[AC+01] Mike Addlesee, Rupert Curwen et al. 'Implementing a sentient computing system'. In: *Computer* 34.8 (2001), pp. 50–56. ISSN: 00189162. DOI: 10.1109/2.940013.

[ACW97] David A. Atchison, W. Neil Charman and Russell L. Woods. 'Subjective Depth-of-Focus of the Eye'. In: *Optometry & Vision Science* 74.7 (1997), pp. 511–520.

[AG+11] Michelle Annett, Tovi Grossman et al. 'Medusa : A Proximity-Aware Multi-touch Tabletop'. In: *UIST '11 Proceedings of the 24th annual ACM symposium on User interface software and technology*. 2011, pp. 337–346. ISBN: 9781450307161.

[AJ+12] M.R. Andersen, T. Jensen et al. *Kinect Depth Sensor Evaluation for Computer Vision Applications*. Tech. rep. Department of Engineering, Aarhus University, Denmark, 2012. DOI: TechnicalReportECE-TR-6.

[Aln15] Mohammed Abdulhamid Alnusayri. 'Proximity Table: Exploring Tabletop Interfaces that Respond to Body Position and Motion'. PhD thesis. 2015.

[AS05] Mark Ashdown and Yoichi Sato. 'Attentive Interfaces for Multiple Monitors'. In: *CHI 2005 Workshop on Distributed Display Environments*. ACM, 2005, pp. 4–5.

[Bae93] Ronald M. Baecker. *Readings in Groupware and Computer-Supported Cooperative Work: assisting human-human collaboration*. Vol. 168. 1993, p. 882. ISBN: 1558602410. DOI: 10.1176/appi.ajp.2011.11.1228.

[BB+10] Sebastian Boring, Dominikus Baur et al. 'Touch projector: mobile interaction through video'. In: *SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*. 2010, pp. 2287–2296. ISBN: 9781605589299. DOI: 10.1145/1753326.1753671.

[BB+14] Gilbert Beyer, Vincent Binder et al. 'The Puppeteer Display : Attracting and Actively Shaping the Audience with an Interactive Public Banner Display'. In: *DIS '14 Proceedings of the Designing Interactive Systems Conference*. 2014, pp. 935–944. ISBN: 9781450329026.

[BB+93] Steve Benford, Adrian Bullock et al. 'From rooms to Cyberspace: models of interaction in large virtual computer spaces'. In: *Interacting with Computers* 5.2 (1993), pp. 217–237. ISSN: 09535438. DOI: 10.1016/0953-5438(93)90019-P.

[BB09] Xiaojun Bi and Ravin Balakrishnan. 'Comparing usage of a large high-resolution display to single or dual desktop displays for daily work'. In: *CHI '09 Proceedings of the 27th international conference on Human factors in computing systems*. ACM Press, 2009, pp. 1005–1014. ISBN: 9781605582467. DOI: 10.1145/1518701.1518855.

[BD11] Andrew Bragdon and Rob DeLine. 'Code space: touch+ air gesture hybrid interactions for supporting developer meetings'. In: *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. 2011, pp. 212–221. ISBN: 9781450308717. DOI: 10.1145/2076354.2076393.

[BDB06] Anastasia Bezerianos, Pierre Dragicevic and Ravin Balakrishnan. 'Mnemonic rendering: an image-based approach for exposing hidden changes in dynamic displays'. In: *Proceedings of the ACM Symposium on User Interface Software and Technology*. 2006, pp. 159–168. ISBN: 1-59593-313-1. DOI: 10.1145/1166253.1166279.

[BE01] V. Bellotti and K. Edwards. 'Intelligibility and accountability: Human considerations in context-aware systems'. In: *Human-Computer Interaction* 16.2 (2001), pp. 193–212. ISSN: 0737-0024. DOI: 10.1207/S15327051HCI16234_05.

[Ben93] Steve Benford. 'A Spatial Model of Interaction in Large Virtual Environments'. In: *Proceedings of the 3rd European Conference on Computer-Supported Cooperative Work (ECSCW'93)*. Springer, 1993, pp. 109–124. ISBN: 0-7923-2447-1. DOI: 10.1007/978-94-011-2094-4.

[BG+02] Patrick Baudisch, Nathaniel Good et al. 'Keeping Things in Context: A Comparative Evaluation of Focus Plus Context Screens, Overviews, and Zooming'. In: *CHI '02 Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*. 4. ACM, 2002, pp. 259–266. ISBN: 1581134533. DOI: 10.1145/503376.503423.

[BG+14] Sebastian Boring, Saul Greenberg et al. 'The Dark Patterns of Proxemic Sensing'. In: *Computer* 47.8 (2014), pp. 56–60. ISSN: 0018-9162. DOI: 10.1109/MC.2014.223.

[BI+04] Harry Brignull, Shahram Izadi et al. 'The introduction of a shared interactive surface into a communal space'. In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work - CSCW '04*. Vol. 6. 3. ACM Press, 2004, pp. 49–58. ISBN: 1581138105. DOI: 10.1145/1031607.1031616.

[BJ88] Christine V. Bullen and Robert R. Johansen. 'Groupware, a key to managing business teams?' 1988.

[BL+14] Frederik Brudy, David Ledo et al. 'Is Anyone Looking? Mitigating Shoulder Surfing on Public Displays through Awareness and Protection'. In: *Proceedings of The International Symposium on Pervasive Displays - PerDis '14*. New York, New York, USA: ACM Press, 2014, pp. 1–6. ISBN: 9781450329521. DOI: 10.1145/2611009.2611028.

[BL00] M. Beaudouin-Lafon. 'Instrumental Interaction: An Interaction Model for Designing Post-WIMP User Interfaces'. In: *CHI '00 Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 2000, pp. 446–453. ISBN: 1581132166. DOI: 10.1145/332040.332473.

[BLA07] Melissa R. Beck, Daniel T. Levin and Bonnie Angelone. 'Change blindness blindness: beliefs about the roles of intention and scene complexity in change detection.' In: *Consciousness and cognition* 16.1 (2007), pp. 31–51. ISSN: 1053-8100. DOI: 10.1016/j.concog.2006.01.003.

[BM+00] Barry Brumitt, Brian Meyers et al. 'EasyLiving: Technologies for Intelligent Environments'. In: *Handheld and ubiquitous computing, LNCS* 1927.Chapter 2 (2000), pp. 12–29. DOI: 10.1007/3-540-39959-3_2.

[BMG10] Till Ballendat, Nicolai Marquardt and Saul Greenberg. 'Proxemic Interaction: Designing for a Proximity and Orientation-Aware Environment'. In: *ITS '10 ACM International Conference on Interactive Tabletops and Surfaces*. ACM, 2010, pp. 121–130. ISBN: 9781450303996. DOI: 10.1145/1936652.1936676.

[BP00] Paramvir Bahl and Venkata N. Padmanabhan. 'RADAR: An in-building RF-based user location and tracking system'. In: *Proceedings of INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies* 2 (2000), pp. 775–784.

[BPB00] Paramvir Bahl, Venkata N. Padmanabhan and Anand Balachandran. *Enhancements to the RADAR user location and tracking system*. Tech. rep. MSR-TR-2000-12. 2000, pp. 775–784.

[Bra98] Gary R. Bradski. 'Computer Vision Face Tracking For Use in a Perceptual User Interface'. In: *Intel Technology Journal* (1998).

[BRB90] K. Ball, D. L. Roenker and J.R. Bruni. 'Developmental Changes in Attention and Visual Search throughout Adulthood'. In: *The Development of Attention: Research and Theory*. 1990, pp. 489–507.

[BS+05] Steve Benford, Holger Schnadelbach et al. 'Expected, sensed, and desired: A framework for designing sensing-based interaction'. In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 12.1 (2005), pp. 3–30. ISSN: 1073-0516. DOI: 10.1145/1057237.1057239.

[BS09] Barry Brumitt and Steven Shafer. 'Better living through Geometry'. In: *Personal and Ubiquitous Computing* 5.1 (2009), pp. 42–45. ISSN: 00278378.

[Bød89] Susanne Bødker. 'A Human Activity Approach to User Interfaces'. In: *Human-Computer Interaction* 4 (1989), pp. 171–195. ISSN: 0737-0024. DOI: 10.1207/s15327051hci0403_1.

[CA00] Andrew D. Christian and Brian L. Avery. 'Speak Out and Annoy Someone : Experiences with Intelligent Kiosks'. In: *CHI '00 Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. Vol. 2. 1. 2000, pp. 313–320. ISBN: 1581132166.

[Cam57] F.W. Campbell. 'The Depth of Field of the Human Eye'. In: *Optica Acta: International Journal of Optics* 4.4 (1957), pp. 157–164. ISSN: 0030-3909. DOI: 10.1080/713826091.

[Cha08] W. Neil Charman. 'The eye in focus: Accommodation and presbyopia'. In: *Clinical and Experimental Optometry* 91.3 (2008), pp. 207–225. ISSN: 08164622. DOI: 10.1111/j.1444-0938.2008.00256.x.

[Cla03] Herbert H. Clark. 'Pointing and placing'. In: *Pointing: Where language, culture, and cognition meet*. 2003, pp. 243–268. ISBN: 0805840141. DOI: 10.4324/9781410607744.

[CM+00] Keith Cheverst, Keith Mitchell et al. 'Sharing (location) context to facilitate collaboration between city visitors'. In: *IMC'00 Workshop on Interactive Applications of Mobile Computing*. 2000.

[CN+03] Elizabeth Churchill, Les Nelson et al. 'The Plasma Poster Network : Posting Multimedia Content in Public Places'. In: *Proceedings of the Ninth IFIP TC13 International Conference on Human-Computer Interaction(INTERACT '03)*. 2003, pp. 599–606. ISBN: 1586033638.

[CS+03] Mary Czerwinski, Greg Smith et al. 'Toward Characterizing the Productivity Benefits of Very Large Displays'. In: *Proceedings of INTERACT '03*. IOS Press, 2003, pp. 9–16.

[CSDS+08] M. Castrillón-Santana, O. Déniz-Suárez et al. 'Face and Facial Feature Detection Evaluation: Performance Evaluation of Public Domain Haar Detectors for Face and Facial Feature Detection'. In: *VISAPP 2008*. 2008, pp. 167–172.

[CW03] Sunny Consolvo and Miriam Walker. 'Using the Experience Sampling Method to Evaluate Ubicomp Applications'. In: *IEEE Pervasive Computing* 2.2 (2003), pp. 24–31.

[CW60] F.W. Campbell and G. Westheimer. 'Dynamics of Accommodation Responses of the Human Eye'. In: *Journal of Physiology* 151.2 (1960), pp. 285–295.

[DAS01] Anind Dey, Gregory Abowd and Daniel Salber. 'A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications'. In: *Human-Computer Interaction* 16.2 (2001), pp. 97–166. ISSN: 0737-0024. DOI: 10.1207/S15327051HCI16234_02.

[DDS07] Heiko Drewes, Alexander De Luca and Albrecht Schmidt. 'Eye-gaze interaction for mobile phones'. In: *Mobility '07 Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology*. Vol. 07. ACM, 2007, pp. 364–371. ISBN: 9781595938190. DOI: 10.1145/1378063.1378122.

[Dey00] Anind K. Dey. 'Providing Architectural Support for Building Context-Aware Applications'. PhD thesis. 2000, p. 188. ISBN: 049301246X. DOI: 10.1207/S15327051HCI16234_02.

[Dey98] Anind K. Dey. 'Context-aware computing: The CyberDesk project'. In: *Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments*. 1998, pp. 51–54.

[DO+04] Joseph DiVita, Richard Obermayer et al. 'Verification of the Change Blindness Phenomenon While Managing Critical Events on a Combat Information Display'. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 46.2 (2004), pp. 205–218. ISSN: 1547-8181. DOI: 10.1518/hfes.46.2.205.37340.

[Dod05] Neil A. Dodgson. 'Autostereoscopic 3D Displays'. In: *Computer* 38.8 (2005), pp. 31–36.

[DR+00] Alan Dix, Tom Rodden et al. 'Exploiting Space and Location as a Design Framework for Interactive Mobile Systems'. In: *ACM Transactions on Computer-Human Interaction* 7.3 (2000), pp. 285–321. ISSN: 10730516. DOI: 10.1145/355324.355325.

[Dua22] A. Duane. 'Studies in Monocular and Binocular Accommodation, with Their Clinical Application.' In: *Transactions of the American Ophthalmological Society* 20 (1922), pp. 132–157. ISSN: 0021-9673.

[DV+05] Connor Dickie, Roel Vertegaal et al. 'eyeLook: Using Attention to Facilitate Mobile Media Consumption'. In: *UIST '05 Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, 2005, pp. 103–106. ISBN: 159593023X. DOI: 10.1145/1095034.1095050.

[EH+13] Ghadeer Eresha, Markus Haring et al. 'Investigating the influence of culture on proxemic behaviors for humanoid robots'. In: *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*. 2013, pp. 430–435. ISBN: 9781479905072. DOI: 10.1109/ROMAN.2013.6628517.

[EV+05] Jerri D. Edwards, David E. Vance et al. 'Reliability and validity of useful field of view test scores as administered by personal computer.' In: *Journal of clinical and experimental neuropsychology* 27.5 (2005), pp. 529–543. ISSN: 1380-3395. DOI: 10.1080/13803390490515432.

[Fit93] George W. Fitzmaurice. 'Situated information spaces and spatially aware palmtop computers'. In: *Communications of the ACM* 36.7 (1993), pp. 39–49.

[FK+07] Toshiyuki Fujine, Yuhji Kikuchi et al. 'Real-life in-home viewing conditions for flat panel displays and statistical characteristics of broadcast video signal'. In: *Japanese Journal of Applied Physics, Part 1: Regular Papers and Short Notes and Review Papers* 46.3 B (2007), pp. 1358–1362. ISSN: 00214922. DOI: 10.1143/JJAP.46.1358.

[Fre08] Edward G. Freedman. 'Coordination of the Eyes and Head during Visual Orienting'. In: *Experimental Brain Research* 190.4 (2008), pp. 369–387. DOI: 10.1007/ss00221-008-1504-8.

[GB+14] Saul Greenberg, Sebastian Boring et al. 'Dark Patterns in Proxemic Interactions: A Critical Perspective'. In: *Proceedings of the 2014 conference on Designing interactive systems*. ACM, 2014, pp. 523–532. ISBN: 9781450329026. DOI: 10.1145/2598510.2598541.

[GB08] Saul Greenberg and Bill Buxton. 'Usability evaluation considered harmful (some of the time)'. In: *CHI '08 Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems*. ACM Press, 2008, pp. 111–120. ISBN: 9781605580111. DOI: 10.1145/1357054.1357074.

[GG00] Carl Gutwin and Saul Greenberg. 'The Mechanics of Collaboration: Developing Low Cost Usability Evaluation Methods for Shared Workspaces'. In: *Proeedings of IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000 (WET ICE 2000)*. 2000, pp. 98–103.

[GHN84] E. Grandjean, W. Hünting and K. Nishiyama. 'Preferred VDT workstation settings, body posture and physical impairments.' In: *Journal of Human Ergology* 15.2 (1984), pp. 99–104.

[GL13] Saul Greenberg and David Ledo. 'Mobile Proxemic Awareness and Control : Exploring the Design Space for Interaction with a Single Appliance'. In: *CHI EA '13: CHI '13 Extended Abstracts on Human Factors in Computing Systems*. 2013, pp. 2831–2832.

[GM+11] Saul Greenberg, Nicolai Marquardt et al. 'Proxemic Interactions: The New Ubicomp?' In: *Interactions* 18.1 (2011), pp. 42–50.

[GP+14a] Juan E. Garrido, Victor M. R. Penichet et al. 'Gaze-based awareness in complex healthcare environments'. In: *Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare*. 2014, pp. 410–413.

[GP+14b] Juan E. Garrido, Victor M.R. Penichet et al. 'AwToolkit: Attention-Aware User Interface Widgets'. In: *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces - AVI '14*. 2014, pp. 9–16. ISBN: 9781450327756. DOI: 10.1145/2598153.2598160.

[Gre91] Saul Greenberg. 'Computer-supported cooperative work and groupware: an introduction to the special issues'. In: *International Journal of Man-Machine Studies* 34.2 (1991), pp. 133–141. ISSN: 00207373. DOI: 10.1016/0020-7373(91)90038-9.

[Gre99] Saul Greenberg. 'Using Digital but Physical Surrogates to Mediate Awareness , Communication and Privacy in Media Spaces'. In: *Personal Technologies* 3.4 (1999), pp. 182–198.

[Gru01] Jonathan Grudin. 'Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use'. In: *CHI '01 Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2001, pp. 458–465. ISBN: 1581133278. DOI: 10.1145/365024.365312.

[Hal66] Edward T. Hall. *The Hidden Dimension*. Anchor books. Doubleday, 1966, pp. 94–94. ISBN: 0385084765.

[HBB02] J. Hightower, B. Brumitt and G. Borriello. 'The location stack: a layered model for location in ubiquitous computing'. In: *Proceedings Fourth IEEE Workshop on Mobile Computing Systems and Applications*. 2002, pp. 22–28. ISBN: 0-7695-1647-5. DOI: 10.1109/MCSA.2002.1017482.

[HD08] Chris Harrison and Anind K. Dey. 'Lean and Zoom: Proximity-Aware User Interface and Content Magnification'. In: *CHI '08 Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*. ACM, 2008, pp. 8–11. ISBN: 9781605580111. DOI: 10.1145/1357054.1357135.

[HH+02] Andy Harter, Andy Hopper et al. 'The Anatomy of a Context-Aware Application'. In: *Wireless Networks* 8.2.3 (2002), pp. 187–197.

[HH11] Chris Harrison and Scott E. Hudson. 'A New Angle on Cheap LCDs: Making Positive Use of Optical Distortion'. In: *UIST '11*. 2011, pp. 537–540. ISBN: 9781450307161.

[HH94] Andy Harter and Andy Hopper. 'A distributed location system for the active office'. In: *IEEE Network* 8.1 (1994), pp. 1–17. ISSN: 0890-8044. DOI: 10.1109/65.260080.

[HK+05] Kirstie Hawkey, Melanie Kellar et al. 'The Proximity Factor: Impact of Distance on Co-located Collaboration'. In: *GROUP '05*. ACM, 2005, pp. 31–40. ISBN: 1595932232.

[HK+14] William Huang, Ye-sheng Kuo et al. 'Opo : A Wearable Sensor for Capturing Face-to-Face Interactions'. In: *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. 2014, pp. 61–75. ISBN: 9781450331432. DOI: 10.1145/2668332.2668338.

[HL+10] Michael Haller, Jakob Leitner et al. 'The NiCE Discussion Room: Integrating Paper and Digital Media to Support Co-Located Group Meetings'. In: *CHI '10 Proceedings of the 28th international conference on Human factors in computing systems*. ACM, 2010, pp. 609–618. ISBN: 9781605589299.

[Hol97] Jack T. Holladay. 'Proper Method for Calculating Average Visual Acuity'. In: *Journal of Refractive Surgery* 13 (1997), pp. 388–391.

[HP+00] Ken Hinckley, Jeff Pierce et al. 'Sensing techniques for mobile interaction'. In: *Proceedings of the 13th annual ACM symposium on User interface software and technology UIST 00*. 2000, pp. 91–100. ISBN: 1581132123. DOI: 10.1145/354401.354417.

[HR+14] Gang Hu, Derek Reilly et al. 'DT-DT : Top-down Human Activity Analysis for Interactive Surface Applications'. In: *ITS '14 ACM International Conference on Interactive Tabletops and Surfaces*. 2014, pp. 167–176. ISBN: 9781450325875.

[Int01] Stephen S. Intille. 'Change Blind Information Display for Ubiquitous Computing Environments'. In: *UbiComp '02 Proceedings of the 4th international conference on Ubiquitous Computing*. London: Springer-Verlag, 2001, pp. 1–16.

[JH12] Mikkel R. Jakobsen and Kasper Hornbaek. 'Proxemics for Information Visualization on Wall-Sized Displays'. In: *Proxemics'12: Workshop in Conjunction with NordiCHI'12*. 2012.

[JLK08] Wendy Ju, Brian A. Lee and Scott R. Klemmer. 'Range: Exploring Implicit Interaction through Electronic Whiteboard Design'. In: *CSCW '08 Proceedings of the 2008 ACM conference on Computer supported cooperative work*. ACM, 2008, pp. 17–26. ISBN: 9781605580074. DOI: 10.1145/1460563.1460569.

[JM+10] Giulio Jacucci, Ann Morrison et al. 'Worlds of Information : Designing for Engagement at a Public Multi-touch Display'. In: *CHI '10 Proceedings of the 28th international conference on Human factors in computing systems*. 2010, pp. 2267–2276. ISBN: 9781605589299.

[Joh88] Robert R. Johansen. *GroupWare: Computer Support for Business Teams*. New York, NY, USA: The Free Press, 1988. ISBN: 0029164915.

[JS+13] Mikkel R. Jakobsen, Yonas Sahlemariam Haile et al. 'Information visualization and proxemics: design opportunities and empirical findings.' In: *IEEE transactions on visualization and computer graphics* 19.12 (2013), pp. 2386–2395. ISSN: 1941-0506. DOI: 10.1109/TVCG.2013.166.

[JS07] Alejandro Jaimes and Nicu Sebe. 'Multimodal human-computer interaction: A survey'. In: *Computer Vision and Image Understanding* 108.1-2 (2007), pp. 116–134. ISSN: 10773142. DOI: 10.1016/j.cviu.2006.10.019.

[KC+04] Russell Kruger, Sheelagh Carpendale et al. 'Roles of orientation in tabletop collaboration: Comprehension, coordination and communication'. In: *Computer Supported Cooperative Work* 13.5-6 (2004), pp. 501–537. ISSN: 09259724. DOI: 10.1007/s10606-004-5062-8.

[KC+12] Seokhwan Kim, Xiang Cao et al. 'Enabling concurrent dual views on common LCD screens'. In: *CHI '12 Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*. ACM, 2012, pp. 2175–2184. ISBN: 9781450310154. DOI: 10.1145/2207676.2208369.

[KG99] Hideaki Kuzuoka and Saul Greenberg. 'Mediating awareness and communication through digital but physical surrogates'. In: *CHI '99 extended abstracts on Human factors in computer systems - CHI '99*. 1999, pp. 11–12. ISBN: 1581131585. DOI: 10.1145/632724.632725.

[KGH85] Myron W. Krueger, Thomas Gionfriddo and Katrin Hinrichsen. 'VIDEOPLACE—an artificial reality'. In: *ACM SIGCHI Bulletin* 16.4 (1985), pp. 35–40. ISSN: 07366906. DOI: 10.1145/1165385.317463.

[KH+00] J. Krumm, S. Harris et al. 'Multi-camera multi-person tracking for EasyLiving'. In: *Proceedings Third IEEE International Workshop on Visual Surveillance*. 2000, pp. 1–8. ISBN: 0-7695-0698-4. DOI: 10.1109/VS.2000.856852.

[KI+98] Yoshinori Kuno, Tomoyuki Ishiyama et al. 'Human Interface Systems Using Intentional and Unintentional Behaviors'. In: *ICMI '98*. 1998.

[KK06] Milton Katz and P. B. Kruger. 'The human eye as an optical system'. In: *Duane's Clinical Ophthalmology.* 15th ed. Lippincott Williams & Wilkins, 2006. Chap. 33.

[KKS10] Dagmar Kern, Milton Keynes and Albrecht Schmidt. 'Gazemarks - Gaze-Based Visual Placeholders to Ease Attention Switching'. In: *CHI '10 Proceedings of the 28th international conference on Human factors in computing systems*. ACM, 2010, pp. 2093–2102. ISBN: 9781605589299. DOI: 10.1145/1753326.1753646.

[Kru77] Myron W. Krueger. 'Responsive environments'. In: *Proceedings of the June 13-16, 1977, national computer conference on - AFIPS '77*. 1977, p. 423. ISBN: 0750605669. DOI: 10.1145/1499402.1499476.

[KSW01] John Krumm, S. Shafer and A. Wilson. 'How a smart environment can use perception'. In: *Workshop on Sensing and Perception for Ubiquitous Computing (part of UbiComp 2001)*. 2001, pp. 1–5.

[Kuu95] Kari Kuutti. 'Activity Theory as a potential framework for human- computer interaction research'. In: *Context and consciousness: Activity theory and human-computer interaction*. MIT Press, 1995, pp. 17–44.

[Led14] David Ledo. 'Remote Control Design for a Ubiquitous Computing Ecology by'. PhD thesis. 2014.

[LGB15] David Ledo, Saul Greenberg and Sebastian Boring. *Proxemic-Aware Controls : Designing Remote Controls for Ubiquitous Computing Ecologies (Research Report 2015-1069-02)*. Tech. rep. University of Calgary, 2015.

[Lin] John Lind. *Exposure*.

[LMM11] Mehdi Latifi, Ali Mobalegh and Elham Mohammadi. 'Movie Subtitles and the Improvement of Listening Comprehension Ability: Does it help?' In: *The Journal of Language Teaching and Learning* 1.2 (2011), pp. 18–29.

[LR+14] Jennifer Long, Mark Rosenfield et al. 'Visual ergonomics standards for contemporary office environments'. In: *Ergonomics Australia* 10.1 (2014), pp. 1–7.

[Mar13] Nicolai Marquardt. 'Proxemic Interactions in Ubiquitous Computing Ecologies'. PhD thesis. 2013. ISBN: 9781450302685. DOI: 10.1145/1979742.1979691.

[MCT09] Erik Murphy-Chutorian and Mohan Manubhai Trivedi. 'Head Pose Estimation in Computer Vision: A Survey'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.4 (2009), pp. 607–626. DOI: 10.1109/TPAMI.2008.106.

[MDM+11] Nicolai Marquardt, Robert Diaz-Marino et al. 'The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies'. In: *Human Factors*. Vol. 11. UIST '11. Department of Computer Science, University of Calgary. ACM, 2011, pp. 315–325.

[MHG12] Nicolai Marquardt, Ken Hinckley and Saul Greenberg. '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*. ACM, 2012, pp. 13–22. ISBN: 9781450315807. DOI: 10.1145/2380116.2380121.

[MI+99] Elizabeth D. Mynatt, Take Igarashi et al. 'Flatland: New Dimensions in Office Whiteboards'. In: *CHI '99 Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1999, pp. 346–353. ISBN: 0201485591.

[MMN99] Susana Marcos, Esther Moreno and Rafael Navarro. 'The depth-of-field of the human eye from objective and subjective measurements'. In: *Vision Research* 39.12 (1999), pp. 2039–2049. ISSN: 00426989. DOI: 10.1016/S0042-6989(98)00317-4.

[MRC07] Tara Matthews, Tye Rattenbury and Scott Carter. 'Defining, Designing, and Evaluating Peripheral Displays - An Analysis Using Activity Theory'. In: *HumanâĂŞComputer Interaction* 22.1 (2007), pp. 221–261. ISSN: 0737-0024. DOI: 10.1080/07370020701307997.

[MSI02] Regan L. Mandryk, Stacey D. Scott and Kori M. Inkpen. 'Display Factors Influencing Co-located Collaboration'. In: *Interactive Poster at ACM Conf. on Computer-Supported Cooperative Work*. 2002.

[MW+12] Jörg Müller, Robert Walter et al. 'Looking glass: a field study on noticing interactivity of a shop window'. In: *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. 2012, pp. 297–306. ISBN: 9781450310154. DOI: 10.1145/2207676.2207718.

[Nar95] Bonnie A Nardi. 'Activity theory and human-computer interaction'. In: *Context and consciousness: Activity theory and human-computer interaction*. MIT Press, 1995, pp. 7–16. ISBN: 0-262-14058-6.

[NN92] Chava Nachmias and David Nachmias. *Research Methods in the Social Sciences*. Vol. 25. 1992, p. 600. ISBN: 9780761923992. DOI: 10.2307/1946631.

[NS+07] Miguel A. Nacenta, Satoshi Sakurai et al. 'E-conic: a Perspective-Aware Interface for Multi-Display Environments'. In: *UIST '07 Proceedings of the 20th annual ACM symposium on User interface software and technology*. New York, New York, USA: ACM, 2007, pp. 279–288. DOI: 10.1145/1294211.1294260.

[OD+12] Mohammad Obaid, Ionut Damian et al. 'Cultural behaviors of virtual agents in an augmented reality environment'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7502 LNAI.2 (2012), pp. 412–418. ISSN: 03029743. DOI: 10.1007/978-3-642-33197-8-42.

[Ols07] Dan R. Jr. Olsen. 'Evaluating User Interface Systems Research'. In: *UIST '07 Proceedings of the 20th annual ACM symposium on User interface software and technology*. 2007, pp. 251–258. DOI: 10.1145/1294211.1294256.

[Ows13] C. Owsley. 'Visual Processing Speed'. In: *Vision Research* 90 (2013), pp. 52–56. ISSN: 15378276. DOI: 10.1016/j.biotechadv.2011.08.021.Secreted. arXiv: NIHMS150003.

[PCB00] Nissanka B Priyantha, Anit Chakraborty and Hari Balakrishnan. 'The Cricket Location-Support System'. In: *Proceedings of the 6th annual international conference on Mobile computing and networking (Mobicom 2000)*. 2000, pp. 32–43. ISBN: 9729910014.

[PK+08] Peter Peltonen, Esko Kurvinen et al. 'It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre'. In: *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*. 2008, pp. 1285–1294. ISBN: 9781605580111. DOI: 10.1145/1357054.1357255.

[PM+93] Elin Pedersen, Kim McCall et al. 'Tivoli: An Electronic Whiteboard for Informal Workgroup Meetings'. In: *INTERCHI ' 93*. ACM, 1993, pp. 391–398. ISBN: 0897915755.

[PR+03] Thorsten Prante, Carsten Röcker et al. 'Hello.Wall - Beyond Ambient Displays'. In: *Adjunct Proceedings of 5th International Conference on Ubiquitous Computing (Ubicomp '03)*. 2003, pp. 277–278.

[PW73] James F. Parker and Vita R. West. *Bioastronautics Data Book*. NASA, 1973.

[RAN05] Matthias Rehm, Elisabeth André and Michael Nischt. 'Lets come together - Social navigation behaviors of virtual and real humans'. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3814 LNAI (2005), p. 336. ISSN: 03029743. DOI: 10.1007/11590323_47.

[RJ+13] Roman Rädle, Hans-christian Jetter et al. 'The Effect of Egocentric Body Movements on Users' Navigation Performance and Spatial Memory in Zoomable User Interfaces'. In: *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*. 2013. ISBN: 9781450322713.

[RL04] Yvonne Rogers and Siân Lindley. 'Collaborating around vertical and horizontal large interactive displays: which way is best?' In: *Interacting with Computers* 16.6 (2004), pp. 1133–1152. ISSN: 09535438. DOI: 10.1016/j.intcom.2004.07.008.

[RLW97] J.M. Rehg, M. Loughlin and K. Waters. 'Vision for a smart kiosk'. In: *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1997, pp. 690–696. ISBN: 0-8186-7822-4. DOI: 10.1109/CVPR.1997.609401.

[RNQ12] Umar Rashid, Miguel a. Nacenta and Aaron Quigley. 'Factors influencing visual attention switch in multi-display user interfaces: A survey'. In: *Proceedings of the 2012 International Symposium on Pervasive Displays - PerDis '12*. 2012, pp. 1–6. ISBN: 9781450314145. DOI: 10.1145/2307798.2307799.

[Rod96] Tom Rodden. 'Populating the Application: A Model of Awareness for Cooperative Applications'. In: *Proceedings of the 1996 ACM conference on Computer supported cooperative work (CSCW '96)*. ACM, 1996, pp. 87–96. ISBN: 0897917650. DOI: 10.1145/240080.240200.

[RS99] Jun Rekimoto and M. Saitoh. 'Augmented surfaces: a spatially continuous work space for hybrid computing environments'. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. 1999, pp. 378–385. ISBN: 0201485591. DOI: 10.1145/302979.303113.

[SAW94] B. Schilit, N. Adams and R. Want. 'Context-aware computing applications'. In: *Workshop on Mobile Computing Systems and Applications*. 1994, pp. 85–90. ISBN: 0-8186-6345-6. DOI: 10.1109/MCSA.1994.512740.

[SB05a] Ben Shneiderman and Benjamin B. Benderson. 'Maintaining Concentration to Achieve Task Completion'. In: *DUX '05 Proceedings of the 2005 conference on Designing for User eXperience*. 2005, pp. 2–7.

[SB05b] Ramona E. Su and Brian P. Bailey. 'Put Them Where? Towards Guidelines for Positioning Large Displays in Interactive Workspaces'. In: *INTERACT 2005 Proceedings of the 2005 IFIP TC13 international conference on Human-Computer Interaction*. 2005, pp. 337–349.

[Sch90] Harvey Richard Schiffman. *Sensation and perception: An integrated approach*. 5th ed. John Wiley & Sons, 1990.

[Sch95] Bill Noah Schilit. 'A System Architecture for Context-Aware Mobile Computing'. PhD thesis. Columbia University, 1995.

[Sco05] Stacey D. Scott. 'Territoriality in Collaborative Tabletop Workspaces'. PhD thesis. University of Calgary, 2005. ISBN: 9788578110796. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1011.1669v3.

[SG+99] Norbert A. Streitz, Jörg Geißler et al. 'i-LAND: an interactive landscape for creativity and innovation'. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 1999, pp. 120–127.

[SGM03] Stacey D. Scott, Karen D. Grant and Regan L. Mandryk. 'System Guidelines for Co-located, Collaborative Work on a Tabletop Display'. In: *Proceeding ECSCW'03 Proceedings of the eighth conference on European Conference on Computer Supported Cooperative Work*. 2003, pp. 159–178. DOI: 10.1007/978-94-010-0068-0_9.

[Sim01] T. Simmons. 'What's the Optimum Computer Display Size?' In: *Ergonomics in Design: The Quarterly of Human Factors Applications* 9.4 (2001), pp. 19–25. ISSN: 1064-8046. DOI: 10.1177/106480460100900405.

[SKB98] Steve Shafer, John Krumm and Barry Brumitt. 'The New EasyLiving Project at Microsoft Research'. In: *Proceedings of the 1998 DARPA/NIST Smart Spaces Workshop*. 1998.

[SMB13] Constantin Schmidt, Jorg Muller and Gilles Bailly. 'Screenfinity : Extending the Perception Area of Content on Very Large Public Displays'. In: *CHI '13: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 1719–1728. ISBN: 9781450318990.

[SP+03] Norbert Streitz, Thorsten Prante et al. 'Ambient Displays and Mobile Devices for the Creation of Social Architectural Spaces'. In: *Puplic and Situated Displays: Social and Interactional Aspects of Shared Display Technologies*. 2003, pp. 387–409. ISBN: 978-1402016776.

[SR+12] Thomas Seifried, Christian Rendl et al. 'Regional Undo / Redo Techniques for Large Interactive Surfaces'. In: *CHI '12 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2012, pp. 2855–2864.

[SSI04] Stacey D. Scott, Carpendale Sheelagh and Kori M. Inkpen. 'Territoriality in collaborative tabletop workspaces'. In: *Proceedings of the 2004 ACM conference on Computer supported cooperative work - CSCW '04*. 2004, pp. 294 –303. ISBN: 1581138105. DOI: 10.1145/1031607.1031655.

[STB07] Garth Shoemaker, Anthony Tang and Kellogg S. Booth. 'Shadow Reaching: A New Perspective on Interaction for Large Wall Displays'. In: *Proceedings of the 20th annual ACM symposium on User interface software and technology*. ACM, 2007, pp. 53–56. ISBN: 9781595936792. DOI: 10.1145/1294211.1294221.

[Tan91] John C. Tang. 'Findings from observational studies of collaborative work'. In: *International Journal of Man-Machine Studies* 34.2 (1991), pp. 143–160. ISSN: 00207373. DOI: 10.1016/0020-7373(91)90039-A.

[Tay05] Gregory Taylor. 'Perceived Processing Strategies of Students Watching Captioned Video'. In: *Foreign Language Annals* 38.3 (2005), pp. 422–427. DOI: 10.1111/j.1944-9720.2005.tb02228.x.

[Tho98] W. David Thomson. 'Eye problems and visual display terminals - The facts and the fallacies'. In: *Ophthalmic and Physiological Optics* 18.2 (1998), pp. 111–119. ISSN: 02755408. DOI: 10.1016/S0275-5408(97)00067-7.

[TQD09] Lucia Terrenghi, Aaron Quigley and Alan Dix. 'A taxonomy for and analysis of multi-person-display ecosystems'. In: *Personal and Ubiquitous Computing* 13.8 (2009), pp. 583–598. ISSN: 1617-4909. DOI: 10.1007/s00779-009-0244-5.

[TT+06] Anthony Tang, Melanie Tory et al. 'Collaborative coupling over tabletop displays'. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*. 2006, pp. 1181–1190. ISBN: 1595933727. DOI: 10.1145/1124772.1124950.

[VB04] Daniel Vogel and Ravin Balakrishnan. 'Interactive public ambient displays: transitioning from implicit to explicit, public to personal, interaction with multiple users'. In: *UIST '04 Proceedings of the 17th annual ACM symposium on User interface software and technology*. ACM, 2004, pp. 137–146. DOI: 10.1145/1029632.1029656.

[VD+02] Roel Vertegaal, Connor Dickie et al. 'Designing Attentive Cell Phones Using Wearable EyeContact Sensors'. In: *CHI '02 extended abstracts on Human factors in computing systems*. ACM, 2002, pp. 646–647. ISBN: 1581134541. DOI: 10.1145/506443.506526.

[Ver14] Jo Vermeulen. 'Designing for Intelligibility and Control in Ubiquitous Computing Environments'. PhD thesis. Hasselt University, 2014.

[VJ04] Paul Viola and Michael J. Jones. 'Robust Real-Time Face Detection'. In: *International Journal of Computer Vision* 57.2 (2004), pp. 137–154. DOI: 10.1023/B:VISI.0000013087.49260.fb.

[VS+06] Roel Vertegaal, Jeffrey S. Shell et al. 'Designing for augmented attention: Towards a framework for attentive user interfaces'. In: *Computers in Human Behavior* 22.4 (2006), pp. 771–789. ISSN: 07475632. DOI: 10.1016/j.chb.2005.12.012.

[Wan12] Miaosen Wang. 'The Proxemic Peddler Framework: Designing a Public Display that Captures and Preserves the Attention of a Passerby'. MSc. University of Calgary, 2012.

[WB10] Andrew D. Wilson and Hrvoje Benko. '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*. New York, New York, USA: ACM Press, 2010, pp. 273–281. ISBN: 9781450302715. DOI: 10.1145/1866029.1866073.

[WB95] Mark Weiser and John Seely Brown. 'Designing Calm Technology'. In: *World Wide Web Internet And Web Information Systems* 1.1 (1995), pp. 1–5.

[WBG12] Miaosen Wang, Sebastian Boring and Saul Greenberg. 'Proxemic Peddler: A Public Advertising Display that Captures and Preserves the Attention of a Passerby'. In: *PerDis '12 Proceedings of the 2012 International Symposium on Pervasive Displays*. ACM, 2012, pp. 3–9. ISBN: 9781450314145. DOI: 10.1145/2307798.2307801.

[WC04] Bin Wang and Kenneth J. Ciuffreda. 'Depth-of-focus of the human eye in the near retinal periphery'. In: *Vision Research* 44.11 (2004), pp. 1115–1125. ISSN: 00426989. DOI: 10.1016/j.visres.2004.01.001.

[WC06] Bin Wang and Kenneth J. Ciuffreda. 'Depth-of-focus of the human eye: Theory and clinical implications'. In: *Survey of Ophthalmology* 51.1 (2006), pp. 75–85. ISSN: 00396257. DOI: 10.1016/j.survophthal.2005.11.003.

[Wei91] Mark Weiser. *The Computer for the 21st Century*. 1991. DOI: 10.1038/scientificamerican0991-94.

[Wer94] T. Wertheim. 'Über die indirekte Sehschärfe'. In: *Zeitschrift für Psychologie und Physiologie der Sinnesorgane* 7 (1894), pp. 172–187.

[WG+07] Dietmar Wüller, Helke Gabele et al. 'The usage of digital cameras as luminance meters'. In: *Electronic Imaging 2007*. International Society for Optics and Photonics, 2007, 65020U1–65020U11.

[WH+92] Roy Want, Andy Hopper et al. 'The Active Badge Location System'. In: *ACM Transactions on Information Systems* 10.1 (1992), pp. 91–102. ISSN: 10468188. DOI: 10.1145/128756.128759.

[WH92] Roy Want and Andy Hopper. 'Active Badges and Personal Interactive Computing Objects'. In: *IEEE Transactions on Consumer Electronics* 38.1 (1992), pp. 10–20.

[WX+06] Shuo Wang, Xiaocao Xiong et al. 'Face-tracking as an augmented input in video games: enhancing presence, role-playing and control'. In: *CHI '06 Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006, pp. 1097–1106. DOI: 10.1145/1124772.1124936.

[WY12] A. B. Watson and J. I. Yellott. 'A unified formula for light-adapted pupil size'. In: *Journal of Vision* 12.10 (2012), pp. 12–12. ISSN: 1534-7362. DOI: 10.1167/12.10.12.

[YK+02] Ming-Hsuan Yang, David J. Kriegman et al. 'Detecting Faces in Images: A Survey'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.1 (2002), pp. 34–58. DOI: 10.1109/34.982883.

[YS+98] Jie Yang, Rainer Stiefelbagen et al. 'Visual Tracking for Multimodal Human Computer Interaction'. In: *CHI '98 Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1998, pp. 140–147. DOI: 10.1145/274644.274666.

[Vis88] Visual Functions Commitee. 'Visual Acuity Measurement Standard'. In: *Italian Journal of Ophthalmology* 2 (1988), pp. 1–18.